

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
1. REPORT DATE (DD-MM-YYYY) 13-01-2011		2. REPORT TYPE Final Technical Report		3. DATES COVERED (From - To) 01-07-2007 to 30-12-2010	
4. TITLE AND SUBTITLE Final Report for "Coordinated Hyperspectral Imaging Nano-Satellite Networks for Space Situational Awareness"				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER FA9550-07-1-0516	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) John McInroy, Suresh Muknahallipatna, Margareta Stefanovic, and Farhad Jafari				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Dept. of Electrical and Computer Engineering University of Wyoming 1000 E. University Ave Laramie, WY 82071				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Air Force Office of Scientific Research				10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-OSR-VA-TR-2012-0480	
12. DISTRIBUTION / AVAILABILITY STATEMENT  A					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT This study has examined how to inspect Resident Space Objects (RSOs) using other Observer satellites, all of which are in fixed orbits. Optimal methods for determining which Observers should characterize each RSO at every sample time have been developed. The resulting inter-satellite measurements are typically at a sufficient range that an affine model of the imaging is highly accurate, so affine transformations have also been studied. Once these measurements are taken, they must then be transmitted to Earth through a communications network, thus congestion control and routing algorithms especially suited for this Space Situational Awareness task have been derived.					
15. SUBJECT TERMS Space Situational Awareness, imaging of satellites, automated planning, satellite communications.					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT  UL	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON John McInroy
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code) 307-766-6137

Coordinated Hyperspectral Imaging Nano-Satellite Networks for  
Space Situational Awareness  
FINAL PROGRESS REPORT  
John E. McInroy, Farhad Jafari, Margareta Stefanovic, and Suresh  
Muknahallipatna  
January 13, 2011  
Air Force Office of Scientific Research  
Grant No. FA9550-07-1-0516  
University of Wyoming

APPROVED FOR PUBLIC RELEASE;  
DISTRIBUTION UNLIMITED.

THE VIEWS, OPINIONS, AND/OR FINDINGS CONTAINED IN  
THIS REPORT ARE THOSE OF THE AUTHORS AND SHOULD  
NOT BE CONSTRUED AS A OFFICIAL DEPARTMENT OF THE  
AIR FORCE POSITION, POLICY, OR DECISION, UNLESS SO  
DESIGNATED BY OTHER DOCUMENTATION.

# Contents

<b>1</b>	<b>Statement of the Problem Studied</b>	<b>4</b>
<b>2</b>	<b>Summary of the Most Important Results</b>	<b>6</b>
2.1	Optimal Observation of Satellites . . . . .	7
2.2	Stable switching congestion control for satellite . . . . .	19
2.2.1	Simulation Results . . . . .	28
2.3	Cross Entropy Accelerated Ant Routing for Satellite . . .	32
2.3.1	Theory . . . . .	34
2.3.2	Simulation Results . . . . .	38
2.4	Satellite Network TCP based Congestion Control . . . .	46
2.4.1	Simulation Results . . . . .	47
2.5	Models for Reconstruction of Images from Multiple Views	56
<b>3</b>	<b>List of Publications</b>	<b>59</b>
<b>4</b>	<b>Scientific Personnel Supported by this Project</b>	<b>59</b>
<b>5</b>	<b>Report of Inventions</b>	<b>60</b>

## List of Figures

1	GEO Orbital Simulation for the first sample time . . . .	10
2	GEO Orbital Simulation after one orbit. . . . .	11
3	GEO Orbital simulation for the last sample time. . . . .	12
4	GEO Binary orbital simulation . . . . .	13
5	The block diagram of ILP-Greedy method. . . . .	14
6	Performance comparison-LEO . . . . .	14
7	Performance comparison-GEO . . . . .	15
8	A snapshot ILP-Greedy planning . . . . .	16
9	Relative orbits of Observers with respect to RSO . . . . .	17
10	Plots of $f(z)$ and $g(z)$ in case 2 . . . . .	25
11	Stable region of $(k_P, k_I)$ . . . . .	27
12	Outputs of Cost-detectable Cost Function (27) . . . . .	28
13	Instant Queue Length using SSC Scheme . . . . .	29
14	Instant Dropping Probability using SSC Scheme . . . . .	29
15	Throughput using SSC Scheme . . . . .	29
16	Congestion Window Size using SSC Scheme . . . . .	30
17	Layout of Iridium constellation . . . . .	34
18	Manhattan Network Topology . . . . .	35
19	Average end to end delay using DV (Scenario 1) . . . . .	39
20	Average end to end delay using LS (Scenario 1) . . . . .	40
21	Average end to end delay using CEARS (Scenario 1) . . . . .	41
22	Average end to end delay using CEAARS (Scenario 1) . . . . .	42
23	Average Buffer Utilization of cluster head . . . . .	49
24	Average Buffer Utilization of middle layer . . . . .	50
25	Average Buffer Utilization of GEO layer . . . . .	51
26	Average Bandwidth Utilization of GEO satellites . . . . .	52
27	Average Buffer Utilization in the GEO layer . . . . .	53
28	Buffer Utilization of GEO Satellites in a Snapshot . . . . .	54
29	Average Bandwidth Utilization of GEO . . . . .	55

# 1 Statement of the Problem Studied

This study has examined how to inspect Resident Space Objects (RSOs) using other Observer satellites, all of which are in fixed orbits. Optimal methods for determining which Observers should characterize each RSO at every sample time have been developed. The resulting inter-satellite measurements are typically at a sufficient range that an affine model of the imaging is highly accurate, so affine transformations have also been studied. Once these measurements are taken, they must then be transmitted to Earth through a communications network, thus congestion control and routing algorithms especially suited for this Space Situational Awareness (SSA) task have been derived. This section will state the problem studied, with further details in the next section.

This study developed methods for regularly characterizing satellites at high resolution and low recurring cost using satellites whose combined orbits collectively have excellent views of space objects. Humans and autonomous spacecraft effectively collaborate by allowing humans to modify and direct both overall objectives and short term goals while using autonomous planning techniques to manage complex, yet well-defined and predictable orbital events and sensor allocations. This allows humans to provide input where they are strong, such as understanding high level goals and reasoning amidst uncertainty. Simultaneously, it allows autonomous optimization and planning techniques to operate in domains where they are strong, in this case when many spacecraft must be jointly controlled over space and time to achieve a clearly defined overall final result. This study formulates observation of multiple satellites by a set of Observer satellites first as a binary integer programming problem. Next, methods are found to relax the formulation to become a convex linear program. Thus global optimality, high speed, and convergence to the optimum are assured. Real-time human input is allowed by modification of the optimization weights. Simulation results confirm the viability of the technique on a variety of orbits, including both low Earth and geosynchronous cases. In addition, the simulations indicate that the binary approximation of the

convex relaxation achieves nearly optimal performance. The method is fully general, encompassing 3-D and elliptical orbits.

Once measurements are taken, they must be transmitted to Earth, so communications networks especially suited for this SSA task have been developed. A stable congestion controller using data-driven, safe switching control theory is developed that improves the dynamic performance of the satellite networks. Robustness is improved by allowing the switching among members of a family of candidate control laws, rather than on using a fixed controller based on *a priori* assumptions about the network dynamics. The stable region of the Proportional-Integral (PI) parameters for a nominal model is explored first. Then, a PI controller, whose parameters are adaptively tuned by switching among members of a given candidate set using observed plant data, is presented and compared with several standard AQM policy examples. A new cost detectable switching law with the interval cost function switching algorithm is developed, which improves the performance and saves the computational cost; it is then compared with a law commonly used in the switching control literature. Finite-gain stability of the system is proven.

In addition to congestion control, an efficient routing algorithm plays a key role in optimizing network resources in a LEO satellite network. A new routing system based on the Cross Entropy optimization method has been developed in this study. The novel on-demand routing method, named Cross Entropy Accelerated Ant Routing System (CEAARS) is applied in simulations to the regular constellation LEO satellite networks. Simulations on an Iridium-like satellite network compare the proposed CEAARS algorithm with two standard approaches to adaptive routing protocols on the Internet: Distance-Vector (DV) and Link-State (LS), as well as with the original Cross Entropy Ant Routing System (CEARS). DV algorithms are based on the distributed Bellman Ford algorithm, and LS algorithms are an implementation of Dijkstra's single source shortest path algorithm. The results show that CEAARS not only remarkably improves the convergence speed of achieving optimal or suboptimal paths, but also reduces

the number of overhead ants (management packets).

From a single constellation of satellites, several layers are needed to transmit the information to Earth. Thus a multi-layered satellite network has also been developed. Due to the dynamic topology of the network, large propagation delays and bulk data transfers, significant delays and loss of data can be encountered with traditional TCP congestion control schemes. To overcome these drawbacks, network snapshots consisting of layered dynamic clusters of satellites are formed with modified TCP congestion control schemes such that the delay and packet losses are at a minimum. Using our new convex optimization formulation, optimal incoming data rates are computed at each satellite irrespective of their layers. They are computed in each snapshot to keep the delay and packet loss at a minimum. During the process of determining optimal incoming data rates a collection of satellites are also identified to perform a particular space situational awareness task.

The resulting inter-satellite measurements are typically at a sufficient range that an affine model of the imaging is highly accurate, so affine transformations and related approximation methods have also been studied. The celebrated Paley-Wiener theorem naturally identifies the spaces of bandlimited functions with subspaces of entire functions of exponential type. This study shows that these spaces remain invariant only under composition with affine maps. After some motivation demonstrating the importance of characterization of range spaces arising from the action of more general composition operators on the spaces of bandlimited functions, the subspaces of  $L_2(\mathbb{R})$  generated by these actions are identified. Extension of these theorems where Paley-Wiener spaces are replaced by the deBranges-Rovnyak spaces are given.

## **2 Summary of the Most Important Results**

A more detailed summary of this study will now be presented in the following subsection.

## **2.1 Optimal Observation of Satellites Using Combined Measurements from Many Networked Observers Cooperatively Controlled by Human and Autonomous Means**

New methods for regularly characterizing satellites at high resolution and low recurring cost using satellites whose combined orbits collectively have excellent views of space objects have been developed. Humans and autonomous spacecraft effectively collaborate by allowing humans to modify and direct both overall objectives and short term goals while using autonomous planning techniques to manage complex, yet well-defined and predictable orbital events and sensor allocations. This allows humans to provide input where they are strong, such as understanding high level goals and reasoning amidst uncertainty. Simultaneously, it allows autonomous optimization and planning techniques to operate in domains where they are strong, in this case when many spacecraft must be jointly controlled over space and time to achieve a clearly defined overall final result. Characterization of satellites in geosynchronous orbit is especially important, as they are too far away for high resolution observation from Earth. This report formulates observation of multiple satellites by a set of Observer satellites first as a binary integer programming problem. Next, methods are found to relax the formulation to become a convex linear program. Thus global optimality, high speed, and convergence to the optimum are assured. Real-time human input is allowed by modification of the optimization weights. For instance, if a human decides that a particular object requires extra observation immediately, then this paper shows how this short term goal can be achieved by weight modification. Similarly, if a human operator desires that an object be observed more persistently and accurately than other objects, then this paper develops weighting methods to achieve this long term goal. Simulation results confirm the viability of the technique on a variety of orbits, including both low Earth and geosynchronous cases. In addition, the simulations indicate that the binary approximation of the convex relaxation achieves nearly optimal performance. The method is fully general, encompassing 3-D



and elliptical orbits.

In order to automatically plan these satellite to satellite observations, we have developed a new relaxed Integer Linear Programming (relaxed-ILP) algorithm which can rapidly generate optimal plans. Fig. 1-4 depict a larger scale simulation ( $n=40$  RSOs,  $m=10$  Observers,  $N=201$  samples) of nearly Geostationary orbits. The number of RSOs and Observers is representative of a true GEO characterization task, as there are approximately 400 active GEO satellites currently, thus this scenario would characterize the most important 10 % of the GEO satellites. The total simulation time covers two orbits. The RSOs are rotating the same direction as the Earth (as is customary), while the Observers rotate in the opposite direction so they will pass nearby all of the RSOs twice per orbit. Fig. 1 illustrates the first sample time—due to the high GEO altitude, occlusion by the Earth is not an issue, and all RSOs are visible and are actively being sensed by some Observer. Each Observer spends most of the sample time on a single RSO but also senses other RSOs for a small portion of the sample time. Note that this result is impractical, as the Observers are attempting to slew towards too many RSOs per sample, and many RSOs are far away. These problems can be partially corrected through a binary approximation which forces each Observer to view only one RSO per sample, but slewing issues become more pronounced as the slewing rates become high, as in the practical retro-GEO case. Fig. 2 depicts the results halfway through the simulation (after a single orbit has been completed). At this point, there is still some marked variability in the quality of each RSO’s characterization (as indicated by the size of the X marks). Fig. 3 illustrates the final results—many observations are taking place simultaneously, and the final results are excellent—all RSOs have been observed nearly equally.

Fig. 4 illustrates the results of the binary approximation, which concentrates all sensing on a single RSO per Observer during each sample time to crudely reduce slewing. In this simulation, it displays high performance that is very close to optimal because all RSOs are of very nearly equal size, which indicates that they have been observed

at nearly equal quality. Note that during the last sample, each Observer sensed either the first or second closest RSO, depending upon the optimal plan.

To improve the binary approximation, the relaxed-ILP algorithm just illustrated has been combined with a Greedy optimization which finds the optimal binary threshold. This new ILP-Greedy algorithm is an amalgam of the relaxed ILP and a Batch Greedy algorithm, and it is based on the block diagram illustrated in Figure 5. As shown, first the ILP solution is found using the ILP relaxation method, then a binary approximation will be done by an optional threshold  $t > 0.5$ .

After the binary approximation process, the plan will not have used all Observers at all sample times because a high threshold sets all elements in those columns to zero. In the last step of this mixed-algorithm, the batch greedy algorithm will be employed to fill these zero columns. Note that when the threshold,  $t$ , equals zeros then a high performance but impractical pure linear program is used, while when  $t = 1$ , the ILP-Greedy algorithm is nothing but a batch greedy algorithm. As the Batch Greedy algorithm is fast, one can run the algorithm with different values of  $t$  and choose the best planning among those to get a practical, but slightly sub-optimal plan. Simulation results show that the mixed algorithm with optimal threshold values always performs better than either the relaxed-ILP or the Batch Greedy algorithms.

Figures 6 and 7 illustrate typical results of all discussed planning methods for two different planning problems. All satellites in the problem related to Figure 6 are arbitrarily chosen within the Low Earth Orbit (LEO), but all satellites related to the other figure orbit in Geosynchronous orbit (GEO).

The planning problem in LEO is a comparatively small problem ( $n = 5$  RSOs,  $m = 3$  Observers,  $N = 500$  Samples and  $L = 2$  side of interests  $\rightarrow 5 \times 3 \times 500 = 7500$  unknowns), while the application in GEO is a much bigger planning problem ( $n = 40$  RSOs,  $m = 8$  Observers,  $N = 901$  Samples and  $L = 2$  side of interests  $\rightarrow 40 \times 8 \times 901 = 288320$  unknowns). In both Figures, black dotted, green dashed and red dashed-dot lines imply the upper bound obtained from ILP-Relaxation

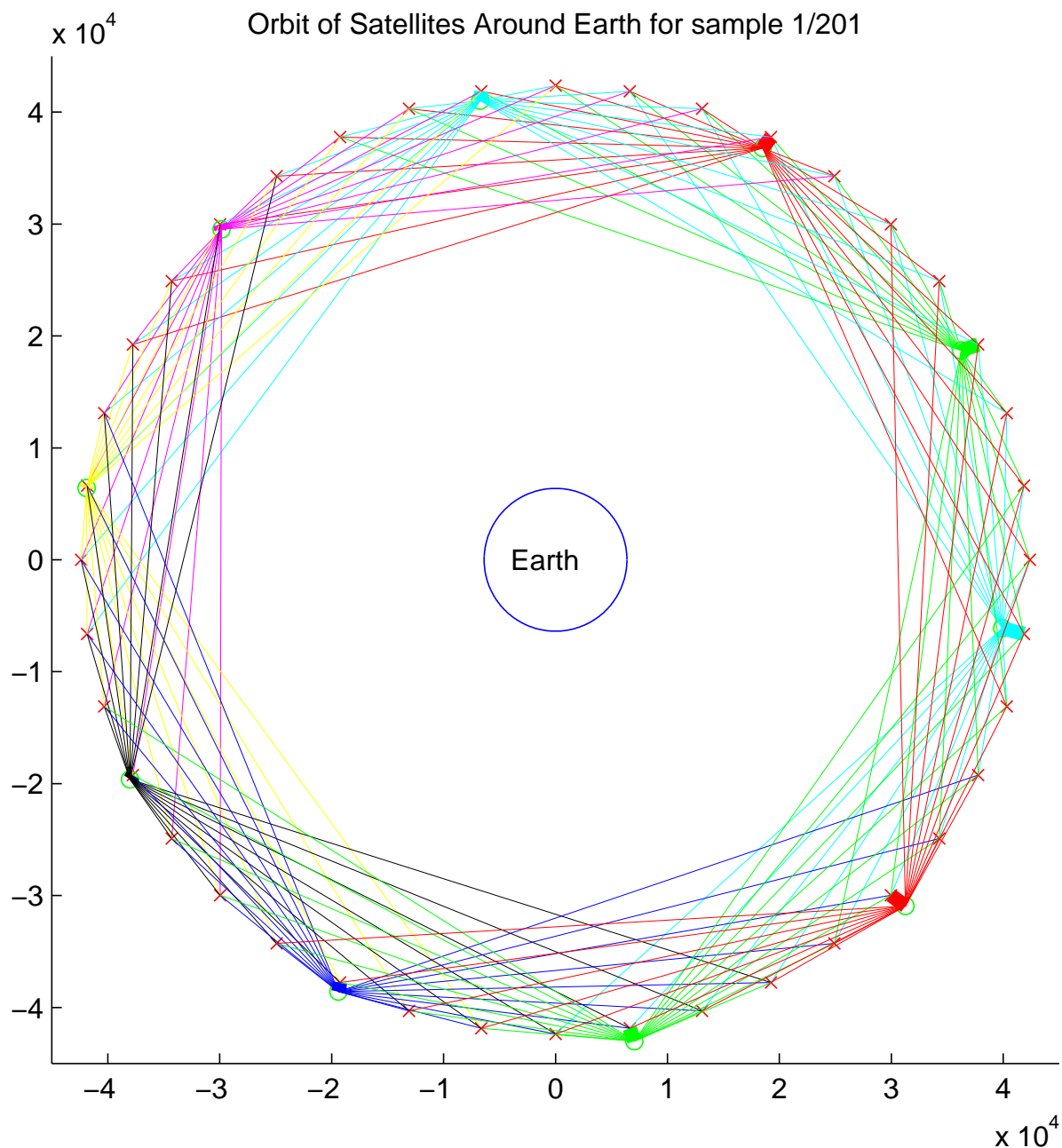


Figure 1: GEO Orbital Simulation for the first sample time. Observers are indicated by green circles and RSOs by red  $x$ 's. The width of a line connecting an Observer to an RSO is proportional to the percentage of that sample time during which the Observer collects data from the corresponding RSO, and the size of the RSO is proportional to the total quality of its observation at the current sample time

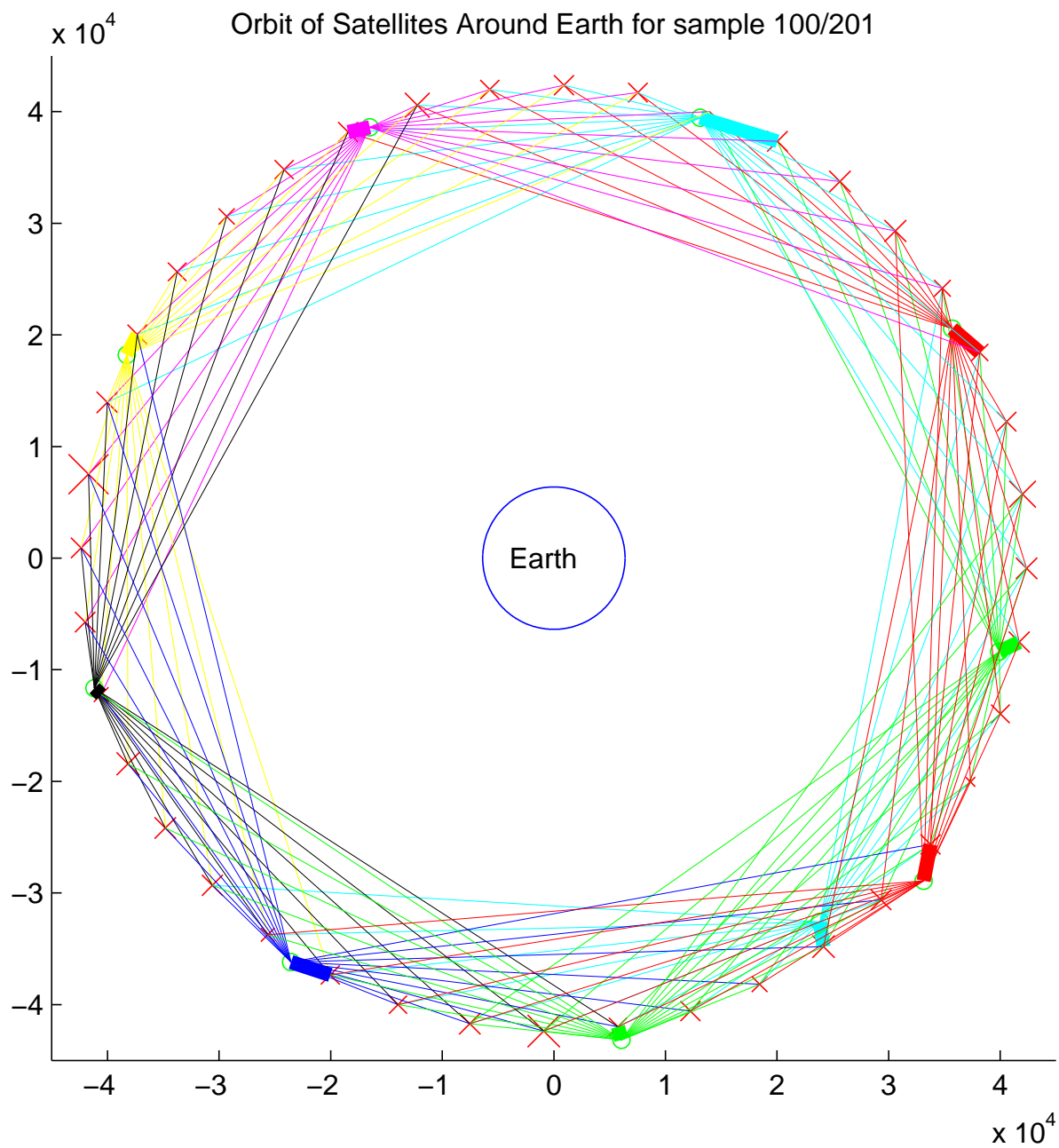


Figure 2: GEO Orbital Simulation after one orbit.

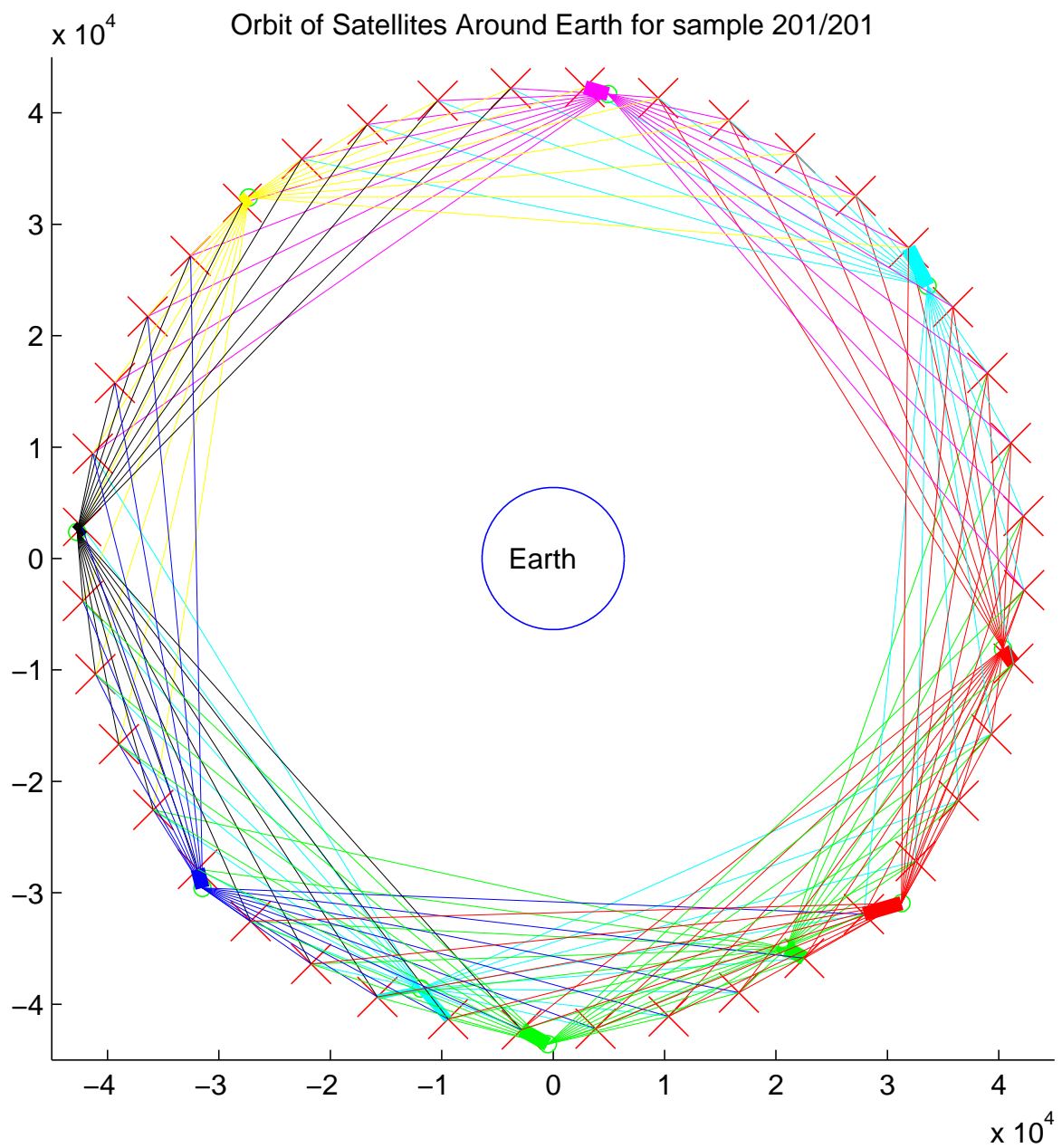


Figure 3: GEO Orbital simulation for the last sample time.

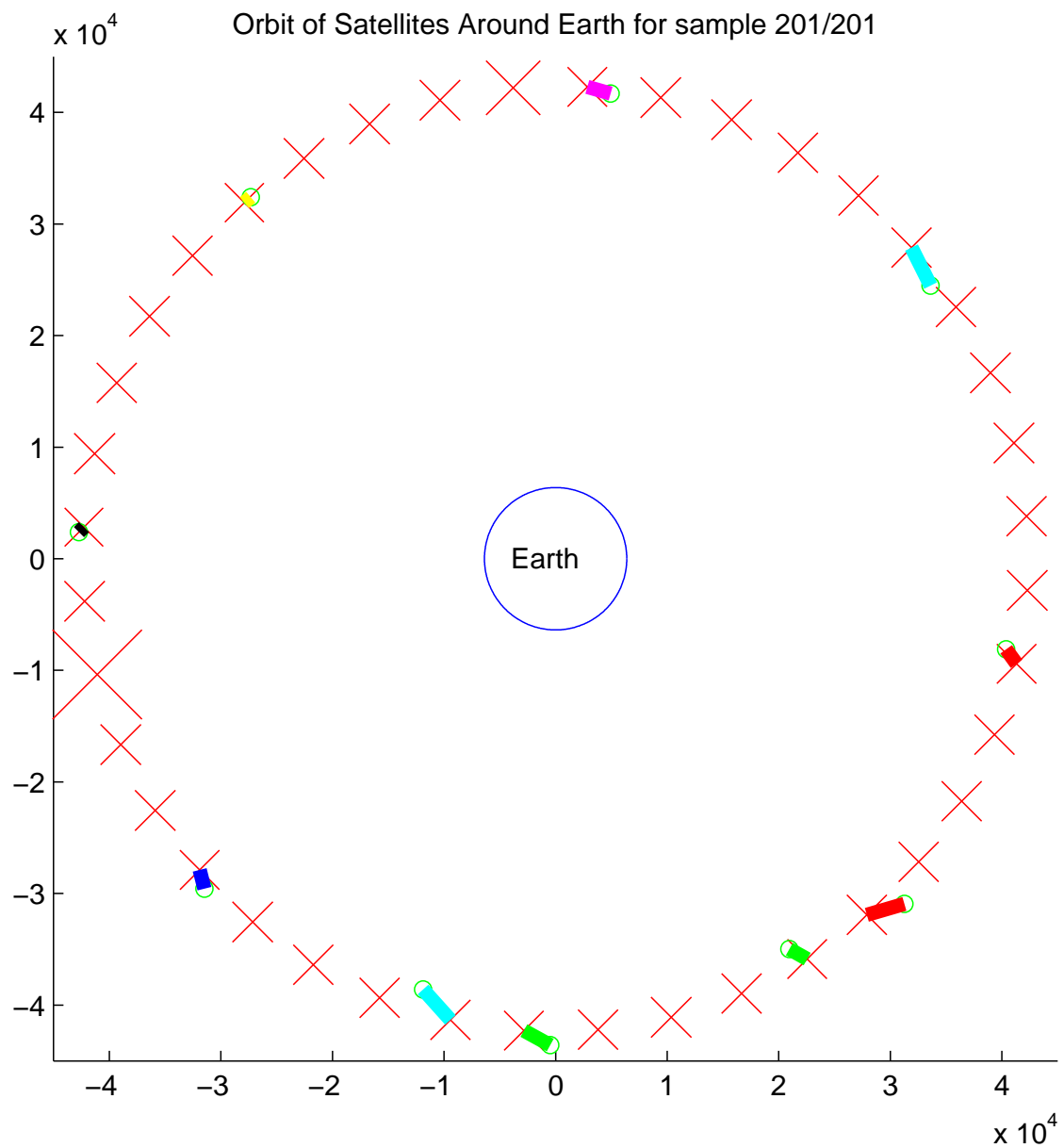


Figure 4: GEO Binary orbital simulation depicted at the last sample time. At this stage the result becomes practical, as all Observers are measuring nearby RSOs, with only one RSO viewed each sample time by each Observer.

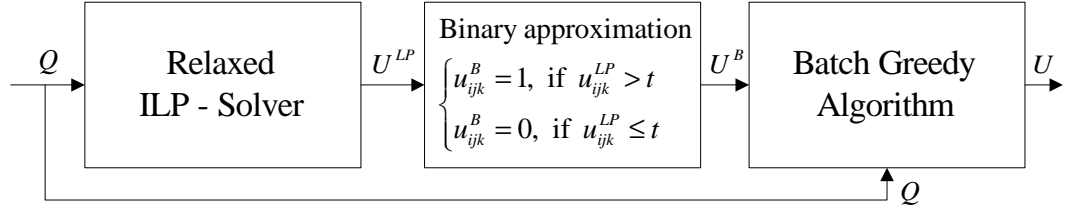


Figure 5: The block diagram of ILP-Greedy method.

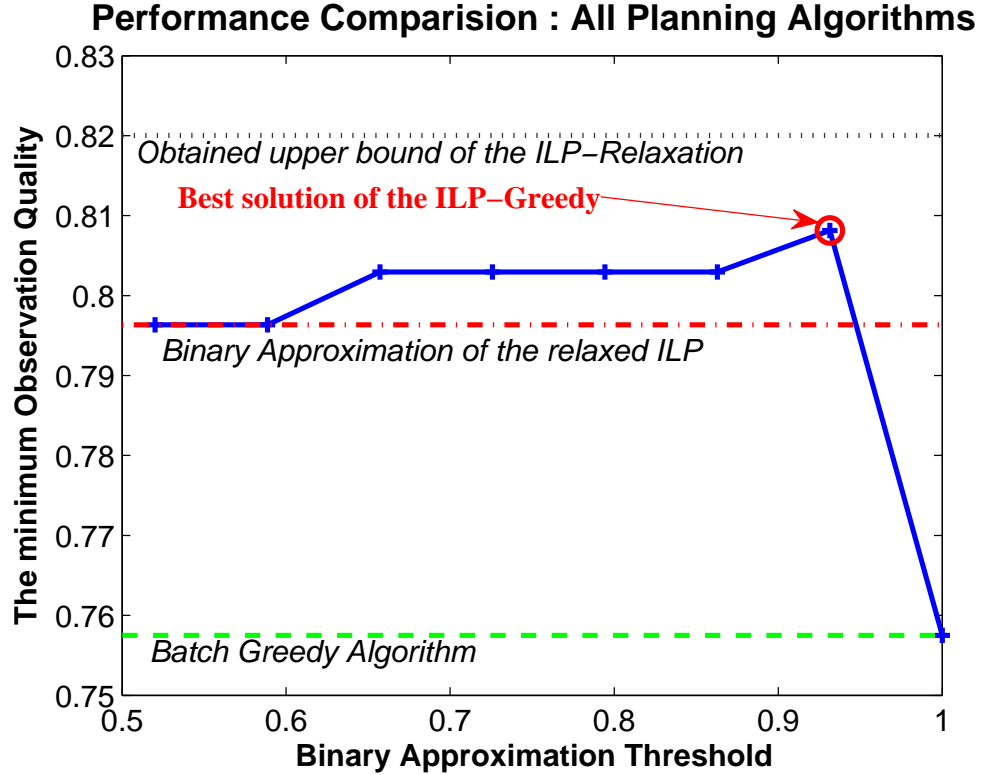


Figure 6: Performance comparison between different planning algorithms for LEO satellites.

method, solution of Batch Greedy algorithm and solution of the Binary approximation of relaxed ILP, respectively. The  $x$  axis is used to show different values of the binary approximation threshold and the  $y$  axis points out the minimum observation quality of all RSOs in all sides of interests. A red circle marker is used to show the best solution of ILP-Greedy method for the different threshold values.

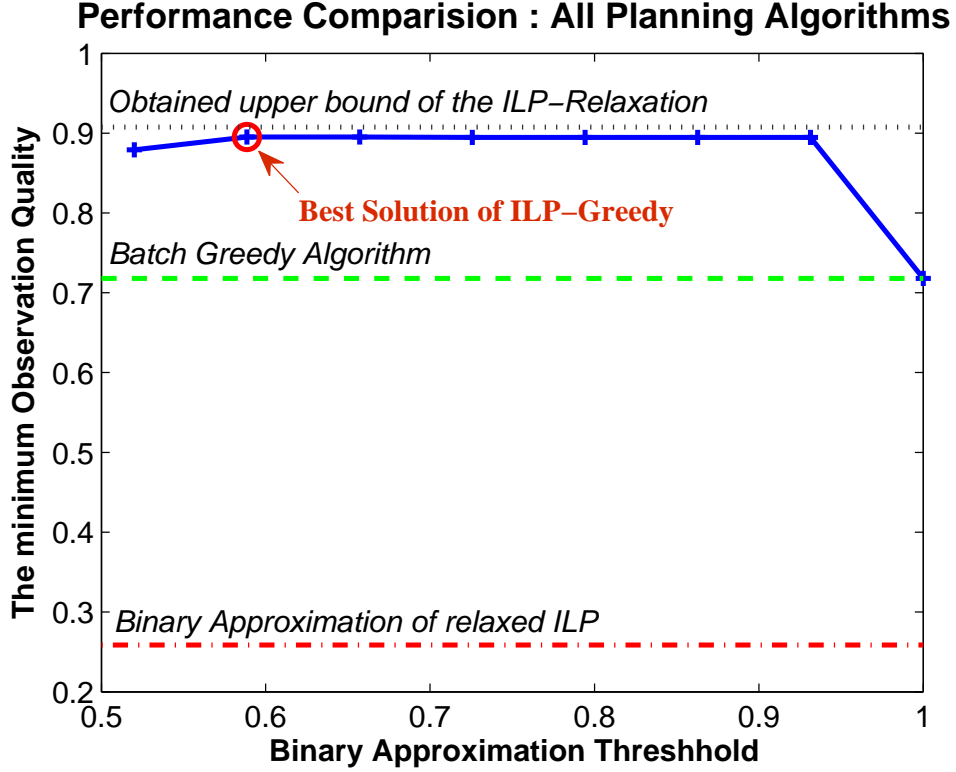


Figure 7: Performance comparison between different planning algorithms for GEO satellites.

The most noticeable feature of the results is that the solution of ILP-Greedy algorithm is closer to the upper bound than the other methods, and it simply means the ILP-Greedy offers a better solution. Moreover, in the LEO problem, the ILP method performs better than the batch Greedy algorithm, but in the GEO problem the converse applies, and it confirms that neither of these two approximate methods is preferable in general. A very powerful feature of this overall approach is that an upper bound on the gap between the found sub-optimal solution and the unknown optimal solution is available. The ILP-relaxation algorithm provides an optimal but physically unrealizable solution. So if performance approaches it ( as it does in both Figures 6 and 7), then the sub-optimal solution is very nearly optimal.

Figure 8 depicts a snapshot of an observation planning for the LEO problem at sample time  $k = 158$ . Observers are indicated by numbered



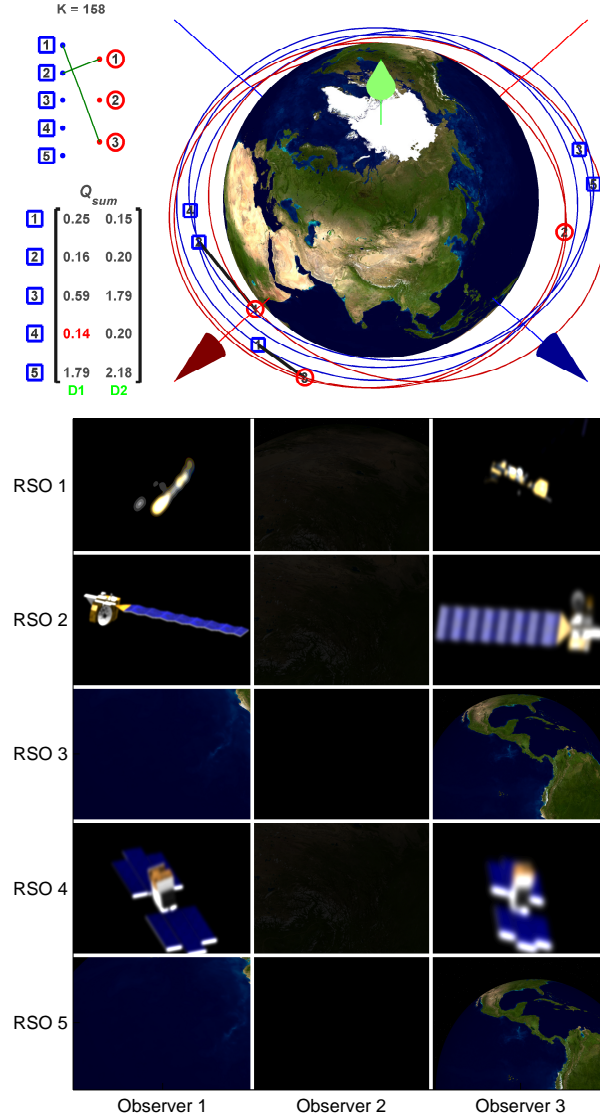


Figure 8: A snapshot ILP-Greedy planning in LEO orbit at sample time  $K = 158$ .

red circles and RSOs by numbered blue squares. All satellites are moving in six different orbits, 3 RSO orbits and 3 Observer orbits. Black lines are utilized to indicate that a RSO is inspected by the connecting observer at this time instance.  $Q_{sum}$  denotes cumulative observation quality matrix, and it shows the observation qualities of all RSOs in every individual side of interest. Cartesian coordinate axes  $x$ ,  $y$  and  $z$  are shown in red, blue and light green respectively. The direction of

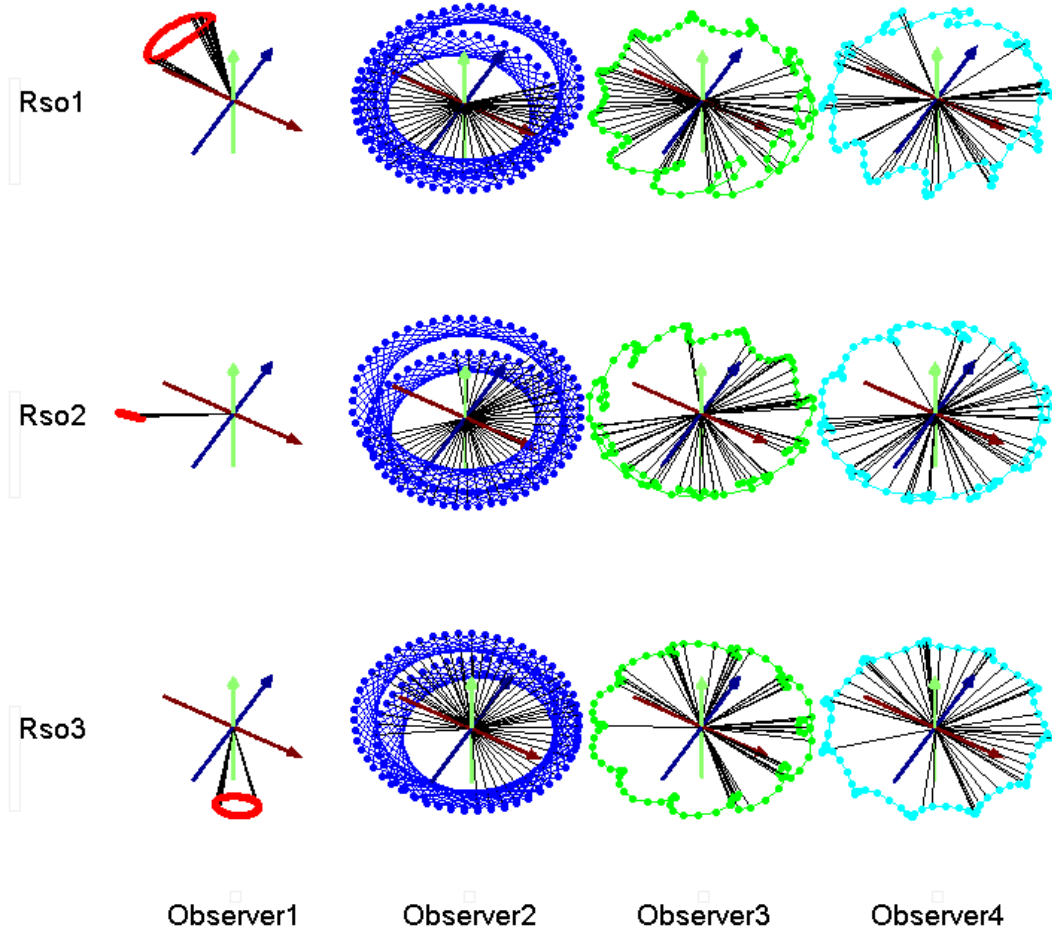


Figure 9: Relative orbits of Observers with respect to RSO

sun light is from positive to negative side of the  $x$  axis.

All possible RSOs images that can be taken from all Observers at the  $k = 158$  time instance are gathered in Figure 8. Every observer has five options (RSOs) to choose for improving the observation quality. The following items explain the reasoning of choosing a RSO for every individual observer. Note that this reasoning is built into the automatic planning system (by minimizing performance indices, not logical inference).

### **Observer 1**

According to Figure 8 RSO 3 and 5 are occluded by the Earth so the observation quality is zero. Although the distance between this observer and RSO 1 is the shortest, due to sun light direction the observer only can inspect the dark side of it. Even though the value of observation quality for RSO 4 is greater than that of RSO 2, the ILP-Greedy scheme chooses RSO 2 because it knows other good opportunities for observing RSO 4 will occur, and the long term planning is possible. In contrast, the Batch-Greedy process lacks this long term perspective, so the Batch Greedy Algorithm would choose RSO 1.

### **Observer 2**

RSO 1 , 2 and 4 are occluded by the Earth. Despite the fact that RSO 3 and 5 are relatively close to observer 1, the deep darkness of that area makes those images worthless.

### **Observer 3**

Again Earth occlusion causes Observer 3 not to see RSO 3 and 5, and RSO 2 and 4 Images are blurred because of the large distance. So, the ILP-Greedy algorithm chooses RSO 1 for inspection. In this case, the Greedy algorithm also chooses RSO 1.

## **Relative Orbit Analysis**

Figure 9 depicts relative orbits of all the Observers with respect to the RSOs in another application within LEO, and it shows that the position of each Observer with respect to each RSO satellite, in a frame attached to, and rotating with, the RSO satellite (Hill's frame). Black lines imply all observations made by observers in different sample times. By analyzing relative orbits of each RSO along with planning, it is easier to evaluate the performance of the proposed method.

## 2.2 Stable switching congestion control for satellite TCP/AQM networks

Once measurements have been taken, they must be eventually transmitted to Earth. Methods for doing so that are suited for this SSA scenario are now outlined.

A stable congestion controller using data-driven, safe switching control theory is developed that improves the dynamic performance of the satellite TCP/AQM networks [1], [2]. Robustness is improved by allowing the switching among members of a family of candidate control laws, rather than on using a fixed controller based on a priori assumptions about the network dynamics. The stable region of the Proportional-Integral (PI) parameters for a nominal model is explored first. Then, a PI controller, whose parameters are adaptively tuned by switching among members of a given candidate set using observed plant data, is presented and compared with several standard AQM policy examples, such as Random Early Detection (RED) and fixed PI control. A new cost detectable switching law with the interval cost function switching algorithm is developed, which improves the performance and saves the computational cost; it is then compared with a law commonly used in the switching control literature. Finite-gain stability of the system is proven. More details follow in the text below.

Congestion on the satellite networks is one of the major communication problems. Transmission Control Protocol (TCP) congestion control algorithms use packet-loss and packet-delay measurements respectively to detect congestion. The current standard in controlling end-to-end congestion in the Internet is the so-called Active Queue Management (AQM); its functionality is based on sensing impending congestion before it occurs and providing feedback information to senders by either dropping or marking packets, so that congestion, causing a significant degradation in network performance, can be avoided.

Prior developments in the robust congestion control considered only a single specific model of the TCP/AQM dynamics, which virtually reduced the effectiveness of the control scheme for all other than the

nominal operating point conditions. The final results of this work build on our previous results in the Safe Switching Control (SSC) of uncertain TCP systems [2], and develop a new cost detectable switching law with the interval cost function switching algorithm, which improves the performance while saving the computational cost; comparisons with a law commonly used in the switching control literature are provided. Finite-gain stability of the system is proven. A fuzzy logic PI controller is proposed as a special candidate to achieve good performance at all nominal points with the available set of candidate controllers (details in [3] and [4]). The new AQM algorithm is applicable to long fat networks and in particular multi-layer satellite networks. Through Matlab and ns-2 simulations, we compare the performance of the new SSC scheme with that of Random Early Detection (RED), adaptive RED, and fixed PI controller. From the simulation results, it is shown that the proposed SSC scheme can adaptively deal with the change of the number of connections and heterogeneous delays.

To design congestion controllers, large scale networks are often simplified according to a ‘dumbbell’ topology, with the network consisting of  $n$  senders, one bottleneck router and one receiver, which in a cluster form of satellite networks correspond to normal satellite nodes, a cluster-head of one cluster and a cluster-head of another (adjacent) cluster, respectively. In this topology, we assume that  $N$  TCP connections represent homogeneous and long lived flows. A fluid-flow model describing the behavior of the average values of key network variables (window size in senders, Round-Trip Time (RTT), and the queue size in the bottleneck router) is given by the following coupled, nonlinear differential equations,

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t - R(t))}{2R(t - R(t))}p(t - R(t)) \quad (1)$$

$$\dot{q}(t) = \frac{W(t)}{R(t)}N(t) - C \quad (2)$$

where  $R(t) = \frac{q(t)}{C} + T_p$  is the RTT,  $W$  is the average TCP window size,  $p$  is the dropping probability of a packet,  $N$  is the number of connections

or TCP sessions,  $C$  is the transmission capacity of the router,  $q$  is the average queue length of the router buffer, and  $T_p$  is the propagation delay. For the design of a nominal plant model, the system is linearized around the equilibrium point  $(W_0, q_0, p_0)$ .

In the prior works, some improvements in the performance (queue utilization, packet drop) have been achieved using adaptive RED and PI schemes; however, all these methods are based on the linearization model at one equilibrium point only, and thus are not effective in the multi-layer satellite networks, where both  $N$  and  $R$  are dynamically varying.

Here we exploit the results on time-delay systems to characterize the stable region of  $k_P$  and  $k_I$  gains. To improve the performance in the dynamically changing operating conditions, we consider first the PI stabilization of first order systems with time delay and extend to the case of TCP AQM. The characteristic equation of dynamic models with time delays has the form:

$$\delta(s) = d(s) + e^{-L_1 s} n_1(s) + e^{-L_2 s} n_2(s) + \cdots + e^{-L_m s} n_m(s) \quad (3)$$

where  $d(s)$  and  $n_i(s)$  for  $i = 1, 2, \dots, m$  are polynomials with real coefficients;  $L_i$  is the time delay;  $\deg[d(s)] = q$  and  $\deg[n_i(s)] \leq q$ ;  $0 < L_1 < L_2 < \cdots < L_m$ . Instead of (3) we can consider the quasi-polynomial  $\delta^*$  as

$$\begin{aligned} \delta^*(s) &= e^{L_m s} \delta(s) \\ &= e^{L_m s} d(s) + e^{(L_m - L_1)s} n_1(s) + \\ &\quad e^{(L_m - L_2)s} n_2(s) + \cdots + n_m(s) \end{aligned} \quad (4)$$

To guarantee the stability of the system, all zeros of the characteristic equation (3) (and thus, all zeros of (4)) must be in the open left half plane. The following theorem gives a necessary and sufficient condition for the stability of (4).

**Theorem 1.** *Rewrite  $\delta^*(s)$  in (4) as*

$$\delta^*(j\omega) = \delta_r(\omega) + j\delta_i(\omega) \quad (5)$$

where  $\delta_r(\omega)$  and  $\delta_i(\omega)$  respectively represent the real and imaginary parts of  $\delta^*(j\omega)$ . Then  $\delta^*(s)$  is stable if and only if:

- 1:  $\delta_r(\omega)$  and  $\delta_i(\omega)$  have only simple, real roots and these interlace.
- 2:  $\delta'_i(\omega_0)\delta_r(\omega_0) - \delta_i(\omega_0)\delta'_r(\omega_0) > 0$  for some  $\omega_0$  in  $(-\infty, \infty)$  where  $\delta'_r(\omega_0)$  and  $\delta'_i(\omega_0)$  denote the first derivative with respect to  $\omega$  of  $\delta_r(\omega)$  and  $\delta_i(\omega)$ , respectively.

The following theorem verifies that the  $\delta_r(\omega)$  and  $\delta_i(\omega)$  have only real roots.

**Theorem 2.** Let  $M$  and  $N$  denote the highest powers of  $s$  and  $e^s$ , respectively in  $\delta^*(s)$ . Let  $\eta$  be an appropriate constant such that the coefficients of terms of highest degree in  $\delta_r(\omega)$  and  $\delta_i(\omega)$  do not vanish at  $\omega = \eta$ . Then for the equations  $\delta_r(\omega) = 0$  or  $\delta_i(\omega) = 0$  to have only real roots, it is necessary and sufficient that in each of the intervals  $-2l\pi + \eta < \omega < 2l\pi + \eta$  with  $l = l_0, l_0 + 1, l_0 + 2, \dots$ ,  $\delta_r(\omega)$  and  $\delta_i(\omega)$  have exactly  $4lN + M$  real roots for a sufficiently large  $l_0$ .

### PI Control for the TCP Plant Dynamics

The linearized TCP plant can be rewritten as the following transfer function:

$$P(s) = \frac{K}{s^2 + a_1s + a_0}e^{-Ls} \quad (6)$$

where  $K$  is the static gain of the plant,  $L$  is the time delay, and  $K, L, a_0$  and  $a_1$  are all nonnegative parameters. Our objective is to analytically determine the set of all stabilizing  $(k_P, k_I)$  values for the given plant. The characteristic equation of the closed loop system is:

$$\delta(s) = K(k_I + k_Ps)e^{-Ls} + (s^2 + a_1s + a_0)s \quad (7)$$

To use Theorem 1 to find the set of stabilizing PI controllers, we deduce the quasi-polynomial  $\delta^*(s)$ , i.e.,

$$\delta^*(s) = \delta(s)e^{Ls} = K(k_I + k_Ps) + (s^2 + a_1s + a_0)se^{Ls} \quad (8)$$

by replacing  $s$  by  $j\omega$ , we get,

$$\delta^*(j\omega) = \delta_r(\omega) + j\delta_i(\omega) \quad (9)$$

By sweeping over all real  $k_P$  and solving a constant gain stabilization problem at each stage, we can determine the region in the  $(k_P, k_I)$  parameter space for which the closed-loop system is stable. We will assume  $k_P > 0, k_I > 0$ . Then, the range of  $k_P$  over which the sweeping needs to be carried out can be narrowed down by using the following theorem:

**Theorem 3.** *Under the above assumptions on  $K, L, a_0$  and  $a_1$ , the range of  $k_P$  values for which a solution exists to the PI stabilization problem of a given open-loop stable plant with transfer function  $P(s)$  as in equation (6) is given by*

$$0 < k_P < \frac{1}{K} \left( a_1 \frac{\alpha}{L} \sin(\alpha) - \cos(\alpha) \left( a_0 - \frac{\alpha^2}{L^2} \right) \right) \quad (10)$$

Where  $\alpha$  is the solution of the equation

$$\tan(\alpha) = \frac{\alpha(2 + a_1 L)}{\alpha^2 - a_0 L^2 - a_1 L} \quad (11)$$

in the interval  $[0, \pi]$ .

*Proof.* By substituting  $z = L\omega$ , the real and imaginary parts of  $\delta^*(s)$  can be rewritten as,

$$\delta_i(z) = \frac{z}{L} \left( K k_P + \left( a_0 - \frac{z^2}{L^2} \right) \cos(z) - a_1 \frac{z}{L} \sin(z) \right) \quad (12)$$

$$\begin{aligned} \delta_r(z) &= K k_I + \left( \frac{z^3}{L^3} - a_0 \frac{z}{L} \right) \sin(z) - a_1 \frac{z^2}{L^2} \cos(z) \\ &= K [k_I - a(z)] \end{aligned} \quad (13)$$

where

$$a(z) = \frac{z}{KL} \left( \sin(z) \left( a_0 - \frac{z^2}{L^2} \right) + a_1 \frac{z}{L} \cos(z) \right) \quad (14)$$

According to Theorem 1, we check two conditions to ensure the stability of the quasi-polynomial  $\delta^*(s)$ .

We start by checking condition 2 of Theorem 1:

$$E(z_0) = \delta'_i(z_0) \delta_r(z_0) - \delta_i(z_0) \delta'_r(z_0) > 0 \quad (15)$$



$$\begin{aligned}\delta'_i(z) = & \frac{Kk_P}{L} - (a_0 + \frac{2a_1z}{L^2} - \frac{z^3}{L^3})\sin(z) + \\ & (\frac{a_0}{L} - \frac{3z^2}{L^3} - a_1\frac{z^2}{L^2})\cos(z)\end{aligned}\quad (16)$$

By substituting  $z_0 = 0$ , we get  $\delta_i(z) = 0$ ,  $\delta'_i(z) = \frac{Kk_P + a_0}{L}$  and  $\delta_r(z) = Kk_I$ . Hence,

$$\begin{aligned}E(z_0) &= \delta'_i(z_0)\delta_r(z_0) \\ &= \frac{Kk_P + a_0}{L}Kk_I > 0\end{aligned}\quad (17)$$

Since  $K > 0$  and  $k_I > 0$ , we have  $k_P > -\frac{a_0}{K}$ . Together with previous assumption, we have  $k_P > 0$  and  $k_I > 0$ .

We now check condition 1 of Theorem 1: the interlacing of the roots of  $\delta_r(z)$  and  $\delta_i(z)$ . From equation (12), we can calculate the roots of the imaginary part, i.e.,  $\delta_i(z) = 0$ . This gives us the following equation

$$z = 0 \quad \text{or} \quad Kk_P + (a_0 - \frac{z^2}{L^2})\cos(z) - a_1\frac{z}{L}\sin(z) = 0 \quad (18)$$

From this we see that one root of the imaginary part is  $z_0 = 0$ . The other roots are difficult to find since we need to solve (18) analytically. However, we can plot the terms involved in (18) and graphically examine the nature of the solution. There are three different cases to consider. In each case, the positive real roots of (18) will be denoted by  $z_j, j = 1, 2, \dots$ , arranged in increasing order of magnitude.

Let  $f(z) = Kk_P + (a_0 - \frac{z^2}{L^2})\cos(z)$  and  $g(z) = a_1\frac{z}{L}\sin(z)$

**Case 1 :**  $k_P = k_u$ . Where  $k_u$  is the maximal value of  $k_P$  so that the graphs of  $f(z)$  and  $g(z)$  are tangent in the interval  $[0, \pi]$ . According to the definition of  $k_u$ , if  $k_P = k_u$ , the curves of  $f(z)$  and  $g(z)$  are tangent at the point  $\alpha$ , i.e.

$$\begin{cases} f(z = \alpha) = g(z = \alpha) \\ f'(z = \alpha) = g'(z = \alpha) \end{cases} \quad (19)$$

From equation group (19), we obtain:

$$\tan(\alpha) = \frac{\alpha(2 + a_1L)}{\alpha^2 - a_0L^2 - a_1L} \quad (20)$$

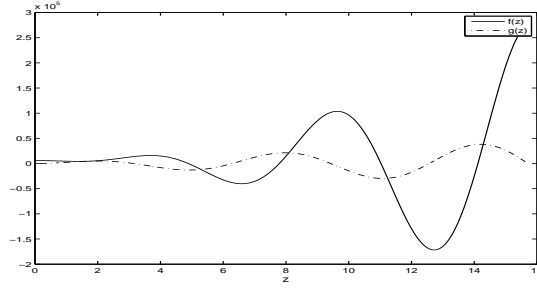


Figure 10: Plots of  $f(z)$  and  $g(z)$  in case 2

Then, the parameter  $k_u$  can be expressed by (21),

$$k_u = \frac{1}{K} \left( a_1 \frac{\alpha}{L} \sin(\alpha) - \cos(\alpha) \left( a_0 - \frac{\alpha^2}{L^2} \right) \right) \quad (21)$$

In this case, we can graph  $f(z)$  and  $g(z)$ . For example, when  $R = 0.03$ ,  $N = 80$ ,  $C = 3750$ , the plot is shown in Fig. 10.

**Case 2 :**  $0 < k_P < k_u$ . In this case, the graph of  $f(z)$  and  $g(z)$  is obtained by moving down the line of  $f(z)$  in Fig. 10.

**Case 3 :**  $k_P > k_u$ . In this case, the graph of  $f(z)$  and  $g(z)$  is obtained by moving up the line of  $f(z)$  in Fig. 10.

Let us now use the results presented in Theorem 2 to check if  $\delta_i(z)$  has only simple roots. By substituting  $s_1 = Ls$  in the expression for  $\delta^*(s)$  in (8), we obtain:

$$\delta^*(s_1) = e^{s_1} \left( \left( \frac{s_1}{L} \right)^3 + a_1 \left( \frac{s_1}{L} \right)^2 + a_0 \frac{s_1}{L} \right) + K \left( k_P \frac{s_1}{L} + k_I \right) \quad (22)$$

We see that for the new quasi-polynomial in  $s_1$ ,  $M = 3$  and  $N = 1$ , where  $M$  and  $N$  designate the highest powers of  $s$  and  $e^s$  which appear in  $\delta^*(s)$ . Next we choose  $\eta = \frac{\pi}{4}$  to satisfy the requirements that  $\delta_r(\eta) \neq 0$  and  $\delta_i(\eta) \neq 0$ . In case 2, we see that, for  $0 < k_P < k_u$ ,  $\delta_i(z)$  has four roots in the interval  $[0, \frac{7\pi}{4}]$ , including a root at the origin. Since  $\delta_i(z)$  is an odd function of  $z$ , it follows that in the interval  $[-\frac{7\pi}{4}, \frac{7\pi}{4}]$ ,  $\delta_i(z)$  has seven roots. Hence,  $\delta_i(z)$  has  $4N + M = 7$  roots in the interval  $[-\frac{7\pi}{4}, \frac{7\pi}{4}]$ , or even  $[-\frac{7\pi}{4}, \frac{9\pi}{4}]$ . Also  $\delta_i(z)$  has two real roots in each of the interval  $[2k\pi + \frac{\pi}{4}, 2(k+1)\pi + \frac{\pi}{4}]$  and  $[-2(k+1)\pi + \frac{\pi}{4}, -2k\pi + \frac{\pi}{4}]$  for  $k = 1, 2, \dots$ . Thus, for  $0 < k_P < k_u$ ,  $\delta_i(z)$  has  $4kN + M$  real roots in the

interval  $[-2k\pi + \frac{\pi}{4}, 2k\pi + \frac{\pi}{4}]$ . Hence, from Theorem 2, we conclude that  $\delta_i(z)$  has only real roots for every  $k_P$  in  $[0, k_u]$ . For the case  $k_P \geq k_u$ , the roots of  $\delta_i(z)$  are not real, thereby ruling out closed-loop stability.  $\square$

We now evaluate  $\delta_r(z)$  at the roots of the imaginary part  $\delta_i(z)$ . For  $z_0 = 0$ , we obtain

$$\begin{aligned}\delta_r(z_0) &= K[k_I - a(0)] \\ &= Kk_I\end{aligned}\tag{23}$$

For  $z_j \neq z_0$  where  $j = 1, 2, 3, \dots$ , using (13) we obtain

$$\delta_r(z_j) = K[k_I - a(z_j)]\tag{24}$$

Interlacing the roots of  $\delta_r(z)$  and  $\delta_i(z)$  is equivalent to  $\delta_r(z_0) > 0$  (since  $k_I > 0$ ),  $\delta_r(z_1) < 0$ ,  $\delta_r(z_2) > 0$ ,  $\delta_r(z_3) < 0$ , and so on. Using this fact and (23, 24) we obtain

$$\begin{aligned}\delta_r(z_1) < 0 &\Rightarrow k_I < a(z_1) \\ \delta_r(z_2) > 0 &\Rightarrow k_I > a(z_2) \\ \delta_r(z_3) < 0 &\Rightarrow k_I < a(z_3) \\ &\vdots\end{aligned}$$

**Lemma 1.** [5]  $a(z_j)$  with odd  $j$  is positive, and the  $a(z_j)$  with even  $j$  is negative.

According to Lemma 1, the range of  $k_I$  becomes

$$0 < k_I < \min_{j=1,3,5,\dots} \{a(z_j)\}\tag{25}$$

The above procedure is summarized in Algorithm 1 (fully presented in the paper [5]).

Applying Algorithm 1 in [5] to the example  $R = 0.03$ ,  $N = 80$ ,  $C = 3750$ , we can get the stable region of  $(k_I, k_P)$  plane is showed in Fig. 11.

**Safe Switching Control System** The unfalsified SSC system, described in [6], utilizes a set of candidate PI controllers, which is one of the most widely used methods in the industry. Fundamentals of the unfalsified control theory and subsequent extension to the SSC algorithms

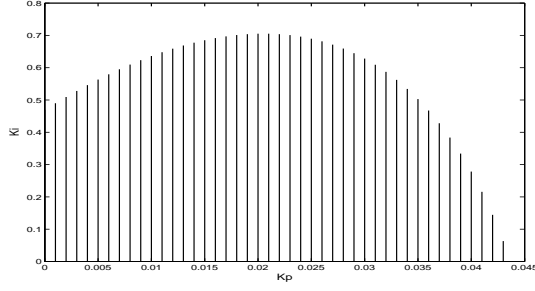


Figure 11: Stable region of  $(k_P, k_I)$

are described in [6] and references therein. To apply this approach, we first construct a bank of PI controllers. Then, we generate the fictitious reference signal and fictitious error signal for each individual PI controller. Given the fictitious reference signal, plant input signal, and plant output signal sets, the “best” (optimal) controller is selected from the candidate set using a properly designed cost function. To improve the overall performance, the candidate controller parameters can be designed off-line using Integral Squared Error (ISE) optimization algorithms, by considering the linearized models in several general cases.

The PI controller dynamics can be expressed as

$$u = (k_P + \frac{k_I}{s})e \quad (26)$$

where  $e = y - r$  is the queue length error signal,  $r$  queue length reference,  $u$  the calculated dropping packet probability, and  $y$  the actual queue length.

### Cost-detectable Cost Function

To account for the plant changes due to varying  $N$  and  $R$ , the following ‘interval cost function’ is proposed:

$$J_i(t) = \max_{\tau \in [t_{n_0}, t]} \frac{\varepsilon \|\Delta u\|_{\tau}^2 + \|\tilde{e}_i\|_{\tau}^2}{\|\tilde{r}_i\|_{\tau}^2 + \alpha} \quad (27)$$

where  $\Delta u$  is the deviation of the dropping probability.  $t_{n_0}$  is the time of the cost function being reactivated at the  $n$ th time. The truncated

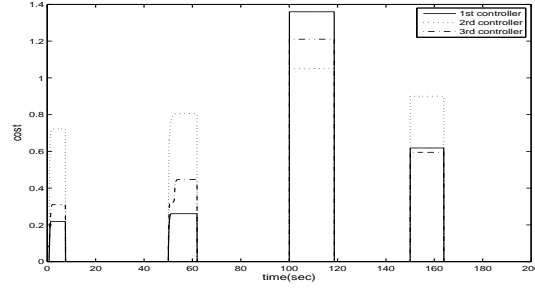


Figure 12: Outputs of Cost-detectable Cost Function (27)

$$\text{2-norm } \|x\|_\tau = \sqrt{\int_{t_{n_0}}^t (x(\tau))^2 d\tau}.$$

---

**Algorithm 1** Interval cost function switching algorithm

---

- 1: Initialization: define a set of candidate controllers, and an initial controller in the loop at the beginning. Set initial cost function output to be 0. Initialize a timer  $T = 0s$ .
  - 2: Measure  $\Delta u$  and  $y$ . Run the timer.
  - 3: Calculate  $\tilde{r}_i$  and  $\tilde{e}_i$ , and  $J_i(t)$ .
  - 4: Switch the controller  $\arg \min_{1 \leq i \leq N} J_i(t)$  into the loop if  $\min_{1 \leq i \leq N} J_i(t) + \epsilon < J_{\text{current\_controller}}(t)$ .
  - 5: Measure  $y$ , and calculate  $e' = r^2 - y^2$ .
  - 6: If  $|e'| > e_{\max}$ , initialize the timer to 0, and go back to step 2.
  - 7: If  $|e'| \leq e_{\max}$  and  $T \leq t_{\max}$ , go back to step 2.
  - 8: If  $|e'| \leq e_{\max}$  and  $T > t_{\max}$ , stop the timer, initialize the cost function output to be 0, and go back to step 5 (shut off the cost monitor).
- 

*Remark* : Shutting the cost monitor in Algorithm 1 saves the computational cost, which is a consideration of high importance in satellite networks.

**Lemma 2.** [5] Consider the cost function in (27) with  $\alpha > 0$ . If the candidate controllers in the set  $\mathbf{K}$  are in PI form, as shown in (26),  $(J, \mathbf{K})$  is cost-detectable

### 2.2.1 Simulation Results

The SSC scheme is simulated using MATLAB and ns-2. The queue length reference  $r = 200 \text{ packets}$ . At the beginning, *Source* : 1 – 30 start sending data to *Destination* : 1; at  $t = 50 \text{ s}$ , *Source* : 1 – 60 switch to send data to *Destination* : 1; at  $t = 100 \text{ s}$ , *Source* : 1 – 60

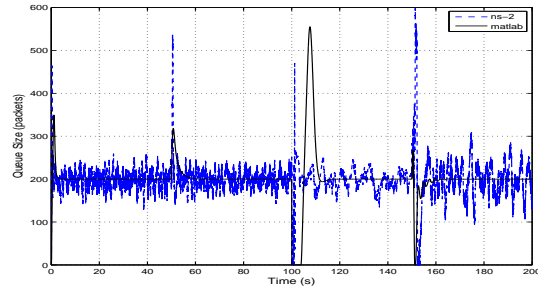


Figure 13: Instant Queue Length using SSC Scheme

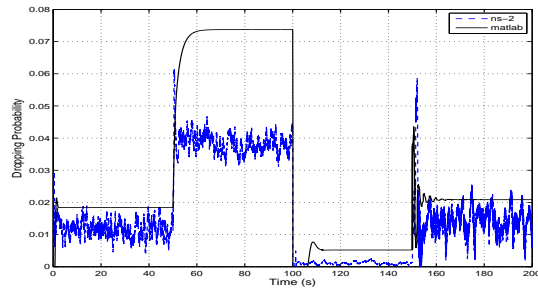


Figure 14: Instant Dropping Probability using SSC Scheme

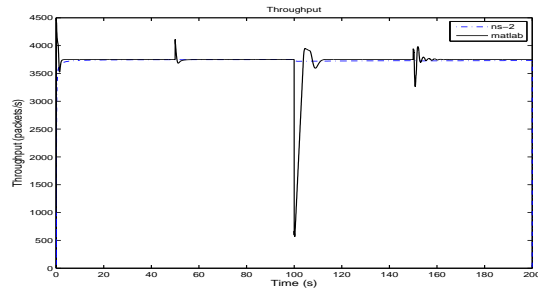


Figure 15: Throughput using SSC Scheme

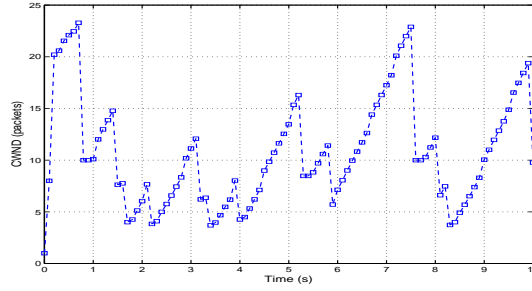


Figure 16: Congestion Window Size using SSC Scheme

switch to send data to *Destination* : 2; at  $t = 150$  s, *Source* : 1 – 120 switch back to send data to *Destination* : 2.

First, based on the linearized models about some nominal points, we find the stable region using Algorithm 1 in [6]. Secondly, we design the candidate controller off-line, using ISE optimization algorithm, as follows. The parameters for the first controller are  $k_P = 0.0001$  and  $k_I = 0.0002$  which are designed around nominal point  $N = 60, R = 0.03$ , those for the second controller are  $k_P = 0.000018$  and  $k_I = 0.00001$  around  $N = 60, R = 0.26$ , and the third controller is a fuzzy one as described above. In Algorithm 1, we assign  $e_{max} = 2000$  packets<sup>2</sup> and  $t_{max} = 5$  s.

Moreover, TCP NewReno scheme is used in the detailed simulation in ns-2, since TCP NewReno is even closer to the fluid model in equation (1), which ignores the slow start procedure. The main contribution of NewReno is the introduction of Fast-Retransmit and Fast-Recovery, which introduces two indications of a packet loss: Retransmission Timeout (RTO) and arrival of three duplicate acknowledgements. If a packet loss is indicated by a timeout, TCP believes that the network is congested and hence enters into the slow start procedure to recover from it. Arrival of three duplicate acknowledgements means that there is still data flowing between the two ends, and congestion does not happen. Hence TCP employs the Fast-Retransmit and Fast-Recovery to avoid reducing the flow abruptly.

We observe that using RED, adaptive RED, or single PI controller does not result in satisfactory tracking the queue length reference. In-

stead, the proposed SSC unfalsified controller can track the reference signal much better.

In Fig. 12, the time intervals when all the cost function outputs drop to 0 signify the case that  $|e'| \leq e_{max}$  and  $T > t_{max}$  (in MATLAB), so the cost monitor is shut off to save the computational cost, and the historical data is discarded to re-initialize the cost level. The change of cost function outputs from zero to nonzero signifies the case that  $|e'| > e_{max}$ , and the current candidate controller might be falsified. Hence, the cost monitoring is reactivated to detect the optimal controller. Fig. 14 shows that the dropping probability increases while the number of connections increases or the propagation delay decreases. Fig. 15 shows that the throughput in the whole period is at or close to  $3750 \text{ packets/s}$ , which means the bandwidth utilization is pretty high. The congestion window size of one TCP flow is zoomed in and showed in Fig. 16, which is consistent with the TCP NewReno scheme.



### 2.3 Cross Entropy Accelerated Ant Routing System for Nano-Satellite Networks

The previous section derived some new congestion control algorithms; this section develops SSA data routing techniques. An efficient routing algorithm plays a key role in optimizing network resources in a LEO satellite network. We have developed a new routing system based on the Cross Entropy optimization method (a heuristic method for solving combinatorial optimization problems) [3], [4]. The novel on-demand routing method, named Cross Entropy Accelerated Ant Routing System (CEAARS) is applied in simulations to the regular constellation LEO satellite networks. By implementing simulations on an Iridium-like satellite network, we compared the proposed CEAARS algorithm with two standard approaches to adaptive routing protocols on the Internet: Distance-Vector (DV) and Link-State (LS), as well as with the original Cross Entropy Ant Routing System (CEARS). DV algorithms are based on the distributed Bellman Ford algorithm, and LS algorithms are an implementation of Dijkstra’s single source shortest path algorithm. The results show that CEAARS not only remarkably improves the convergence speed of achieving optimal or suboptimal paths, but also reduces the number of overhead ants (management packets).

Since the distance between satellite planes changes with the movement of the satellites, the *interplane* intersatellite links (ISLs) are longest when satellites are over the equator and shortest when they are over the polar region boundaries. In the near polar (or Walker star) constellations, such as Iridium and Teledesic, *interplane* ISLs are not maintained as satellites come close to the poles, due to adverse pointing and tracking conditions, and are reestablished when satellites move to lower latitudes. Since the satellite movements cause changes in the network topology, more robust routing algorithms for satellite networks are needed, which can establish and maintain an optimal path between source and destination in a dynamically varying topological setting.

Prior results in the routing algorithms for satellite networks mainly focus on the static snapshot path setup and snapshot transition phases.

Satellites forward the packets according to their routing tables, which are generated in a ground switch centrally, and the routing tables are updated according to a switching table, as the satellite network topology changes. In the above schemes, adaptivity and robustness of a satellite networks are not optimal. Moreover, high energy consumption is required to maintain all snapshot routes subject to the variable topology requirements, which is inappropriate for satellite networks (especially nano-satellite networks with strict energy constraint).

In the present work, a robust Cross Entropy Ant Routing System(CEARS) and its improved version Cross Entropy Accelerated Ant Routing System(CEAARS) are introduced for regular constellations LEO satellite networks (such as near polar -or Walker star- constellations), and compared with the distance-vector (DV) and link-state (LS), which are two known approaches to adaptive routing protocols on the Internet Protocol (IP) networks. Both CEARS and CEAARS are adaptive to the change of network topology, and also robust to the path recovery due to the link and node failure. CEARS is based on the global random search, so it takes a much longer time to find an optimal or suboptimal path at the beginning, and consumes more energy. Due to the capability of sensing the direction of destination, CEAARS is characterized by a faster convergence speed and better energy consumption.

An Iridium-like constellation, shown in Fig. 17, which consists of 66 satellites with 6 planes of 11 satellites per plane in a near polar LEO, is considered as an example. Assume that each satellite or node is capable of having four links, comprising two *intraplane* ISLs forward and back and two interplane ISLs to the satellites in the left and right neighboring planes. However, *cross seam* ISLs are inactive, and *interplane* ISLs are inactive in the near-polar region. This topology can be considered as a variation of the bidirectional seamless Manhattan network, as shown in Fig. 18.

By examining the number of paths, we show in [3] that it is a rare event to find an optimal path by doing a random walk.

In the first phase of the Cross Entropy Ant Routing System (CEARS)

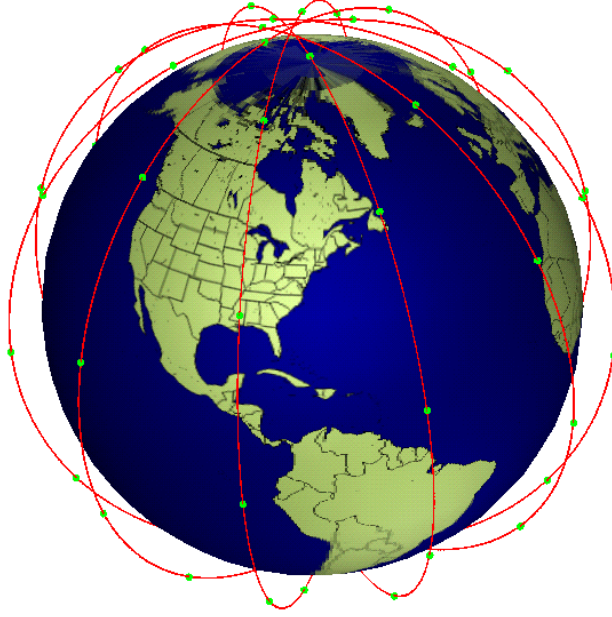


Figure 17: Layout of Iridium constellation

method, ants (or agents) walk randomly through the network from the source node to the destination node, measure the cost of the paths and deposit pheromone to represent the quality of the traversed paths. In the second phase, according to the cost of the discovered paths, the cross entropy between the density of the cost of paths and the density of the routing probabilities is minimized to generate the optimal (lowest cost) path, and the routing probability matrix is updated accordingly.

### 2.3.1 Theory

#### **‘Close to Destination’ Neighbor Nodes Set.**

The following simulation results show that CEARS (which does not take into account the relationship among the locations of the satellite nodes) takes a relatively long time to converge to the optimal paths. This makes the CEARS not suitable as the routing algorithm for short-time connections. Considering the fact that the satellite nodes have certain relation with each other in regular constellations, we want to

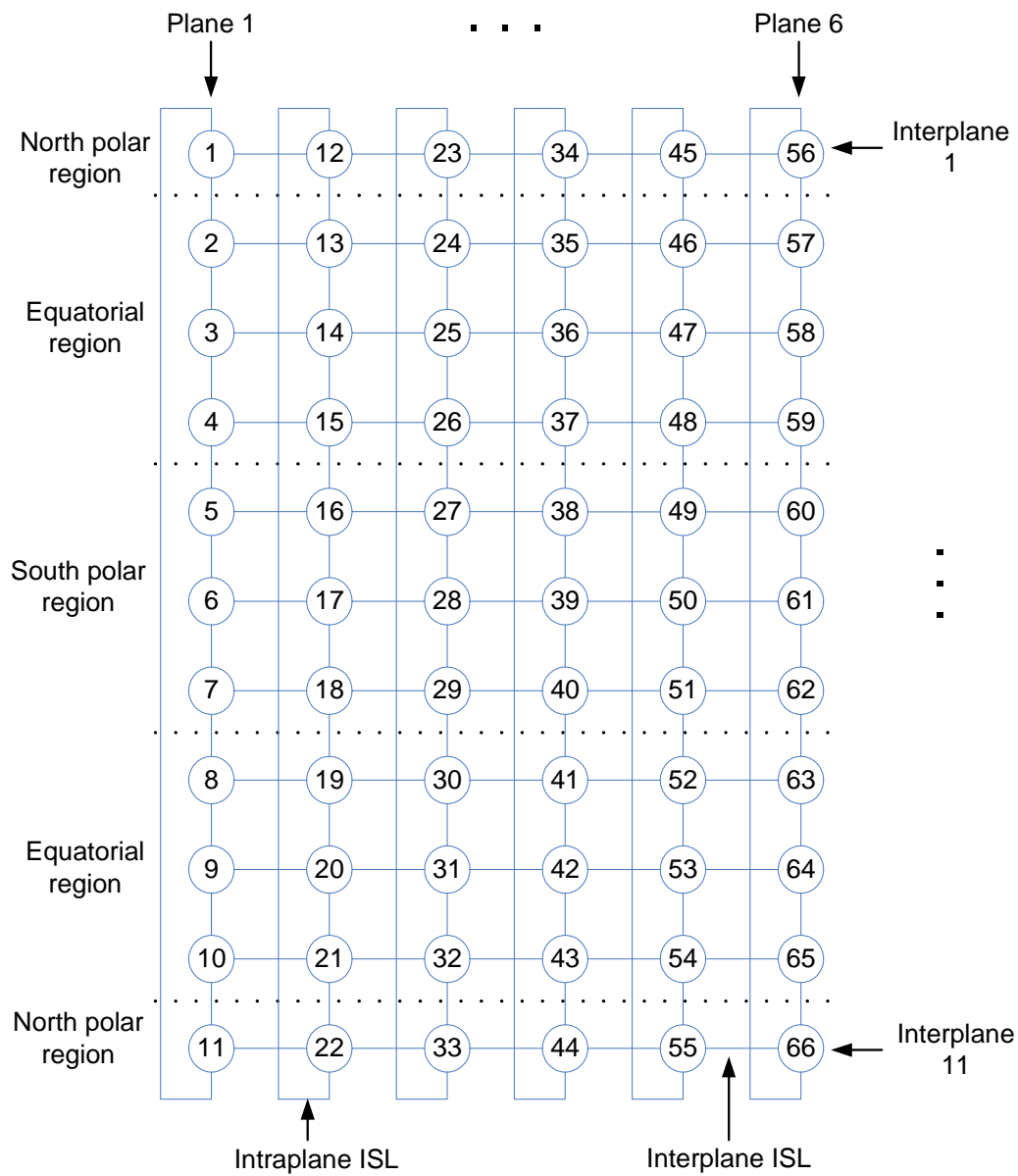


Figure 18: Manhattan Network Topology

improve the convergence speed by adding the so-called *destination-oriented* ants, whose neighbor set includes only close-to-destination neighbor nodes.

The close-to-destination neighbor nodes set is a set in which the nodes are contained in the minimum hop path. To determine the close-to-destination neighbor nodes set with minimum hop metrics, we use the location relations between the current satellite  $n_C$  and the destination satellite  $n_D$  as shown in the pseudo code in [4], in which the neighbor nodes set is narrowed down by comparing the row and column indices of the current satellite with those of the destination satellite.

The CEAARS algorithm is described as follows.

1. A uniformly distributed probability matrix is initialized. Three types of ants: destination-oriented ants, normal ants and explorer ants, are generated at the source node and start to search forward paths to the destination node.
2. For the destination-oriented ants, according to the relative position between the current node and destination node by comparing their row and column indexes, a set of “close-to-destination” neighbor nodes are generated. For other types of ants, the neighbor set includes all neighbor nodes.
3. For normal ants and destination-oriented ants, the next hop node to be visited is selected according to a probability distribution in their own neighbor sets. If no nodes were available in the set of “close-to-destination” neighbor nodes for destination-oriented ants, the set with all neighbor nodes will be considered. For explorer ants, the next hop node to be visited is selected randomly.
4. If the selected node has been visited, it will be avoided.
5. If the forward ants does not arrive at the destination, go to (2).
6. If the forward ants arrived at the destination, the cost of the path is calculated, and the temperature is updated.
7. Backward ants are generated and return on the reserved path, updating the pheromone values and probability distribution values.

**Fuzzy Adaptive Ant Generation Rates** In CEARS, a relatively high ant generation rate is necessary to maintain up-to-date statistics of the network status. This might create significant overhead in terms of the routing traffic, increasing the cost and having a negative impact on the overall network performance. To save computation and bandwidth cost in a satellite network, ant generation rates are adaptively changed

according to the routing convergence condition. For example, at the beginning, all forward ants visit the next hop node randomly, and the backward ants follow their own forward paths reversely and update the pheromone values. During this period, the backward rate is much lower than the forward rate. In this case, we want to assign a high ant generation rate to find the optimal paths as soon as possible. Then, more ants follow the lowest cost paths forward and backward, and the backward rate becomes close to the forward rate. In this case, we want to assign a low ant generation rate to save computation and bandwidth cost. Even though we know how to control the ant generation rate by using human heuristic knowledge, it is difficult to use formal analysis to calculate the explicit ant generation rate. Hence we adopt fuzzy logic, which is easy to use and very flexible.

For the convenience of calculation and satisfactory accuracy, we divide  $r_{bf}$ ,  $\Delta C$ ,  $r_{ng}$ ,  $r_{sg}$ ,  $r_{eg}$ , and  $\beta$  into 2 fuzzy sets: Big (B) and Small (S), where  $r_{ng}, r_{sg}$  and  $r_{eg}$  represent normal ants, destination-oriented ants and explorer ants generation rates respectively.  $\Delta C$  is the difference between the average cost and a pre-set reference cost. In this work, the non-optimal actual cost is usually much greater than the theoretical minimum cost (more than ten times greater). Using the fuzzy logic rules, we can define the reference cost to be three times the minimum cost, and consider all lower cost as 'SMALL', otherwise 'BIG'. By  $r_f(k)$  and  $r_b(k)$  we denote forward rate and backward rate at visit  $k$ , respectively, and  $r_{bf} = \frac{r_b(k)}{r_f(k)}$ . Then we can use the heuristic knowledge about the relationship between forward rate and backward rate to design some simple fuzzy rules for the ant generation rates and evaporation rate shown in Tables 1 and 2.

The control rules are developed with  $r_{bf}$  and  $\Delta C$  as premise variables, and  $r_{ng}$ ,  $r_{sg}$ ,  $r_{eg}$ , and  $\beta$  as consequent variables of the each rule. The structure of the fuzzy rule is described as:

IF  $r_{bf}$  is SMALL and  $\Delta C$  is BIG,  
 THEN  $r_{ng}$  and  $r_{sg}$  are BIG,  $r_{eg}$  is SMALL,  $\beta$  is BIG.

In this work, the corresponding fuzzy sets for  $r_{bf}$ ,  $\Delta C$ ,  $r_{ng}$ ,  $r_{sg}$ ,  $r_{eg}$  and  $\beta$  are specifically  $\{0.3, 0.6\}$ ,  $\{0.05, 0.1\}$ ,  $\{40, 100\}$ ,  $\{40, 100\}$ ,  $\{5, 20\}$ , and  $\{0.6, 0.95\}$ . The normal triangular membership function is used. For the defuzzification procedure that produces a crisp control output from the result of the inference, the Center of Gravity method is used here to get the final output value.

Table 1: Rule base for ant generation rates

$r_{ng} \text{ or } r_{sg}$		$r_{bf}$	
		S	B
$\Delta C$	S	S	S
	B	B	S

$r_{eg}$		$r_{bf}$	
		S	B
$\Delta C$	S	S	S
	B	S	B

Table 2: Rule base for the evaporation rate

$\beta$		$r_{bf}$	
		S	B
$\Delta C$	S	B	B
	B	B	S

In order to play down the impact of the bursty traffic on the forward and backward ants rates, an exponentially weighted moving average (EWMA) low pass filter is designed to estimate the rates as:

$$r_f(k) \leftarrow W_r r_f(k-1) + (1 - W_r) r_g(k-1)$$

$$r_b(k) \leftarrow W_r r_b(k-1) + (1 - W_r) \frac{n_k}{\tau}$$

where  $W_r$  is the time constant for the low pass filter,  $r_b(0) = 0$ , and  $n_k$  is the number of backward ants in time  $[(k-1)\tau, k\tau]$ .

### 2.3.2 Simulation Results

For performance evaluation of the CEAARS, simulation experiments on an Iridium-like topology network are studied on the Network Simulator 2 (ns-2), a discrete event simulator. The following assumptions are considered: the propagation delay of each ISL is  $15ms$ , and all interplane ISLs in one interplane become inactive simultaneously in the

Table 3: Interplane ISLs status on different Snapshots

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$
$I_1$	0	0	0	0	1	1	1	1	1	1	0
$I_2$	1	0	0	0	0	0	1	1	1	1	1
$I_3$	1	1	1	0	0	0	0	0	1	1	1
$I_4$	1	1	1	1	1	0	0	0	0	0	1
$I_5$	0	1	1	1	1	1	1	0	0	0	0
$I_6$	0	0	0	1	1	1	1	1	1	0	0
$I_7$	0	0	0	0	0	1	1	1	1	1	1
$I_8$	1	1	0	0	0	0	0	1	1	1	1
$I_9$	1	1	1	1	0	0	0	0	0	1	1
$I_{10}$	1	1	1	1	1	1	0	0	0	0	0
$I_{11}$	0	0	1	1	1	1	1	1	0	0	0

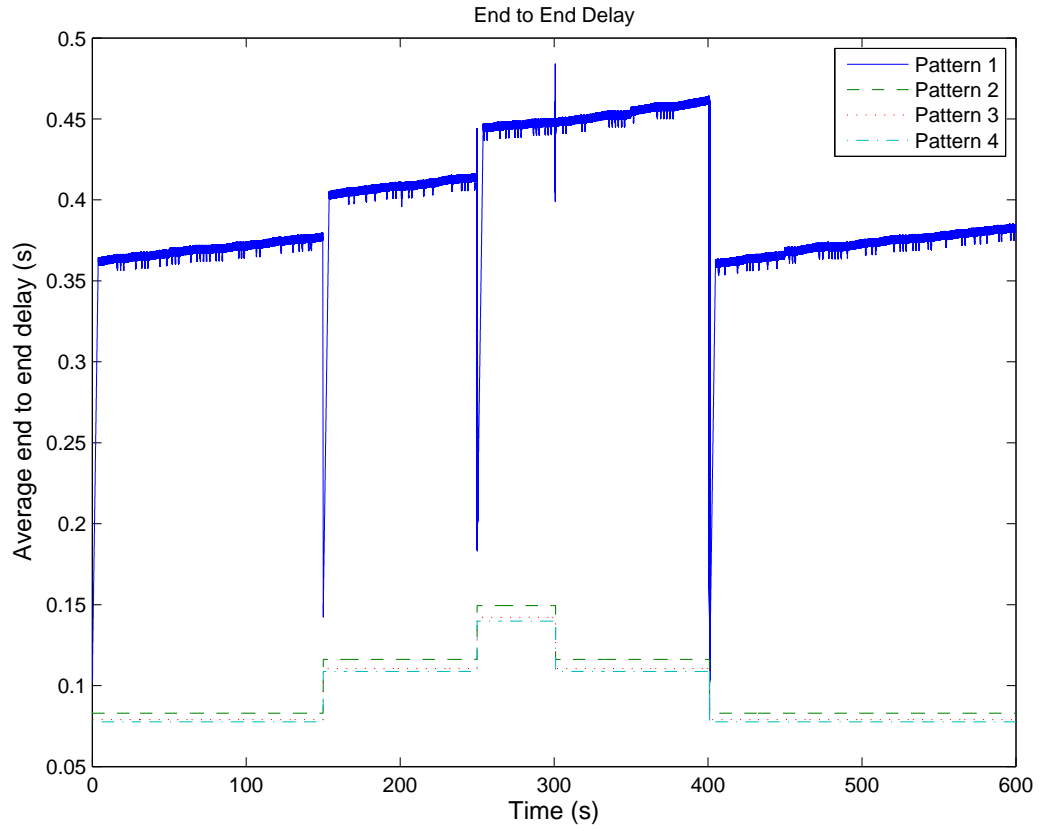


Figure 19: Average end to end delay using DV (Scenario 1)



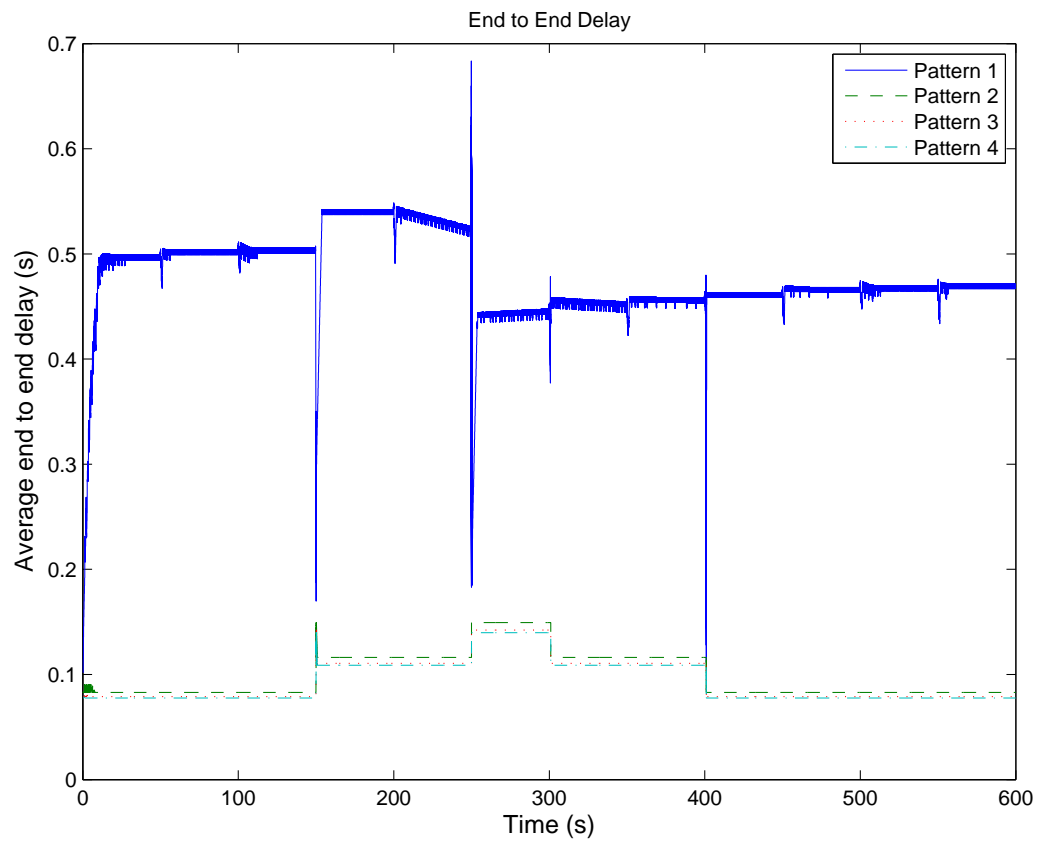


Figure 20: Average end to end delay using LS (Scenario 1)

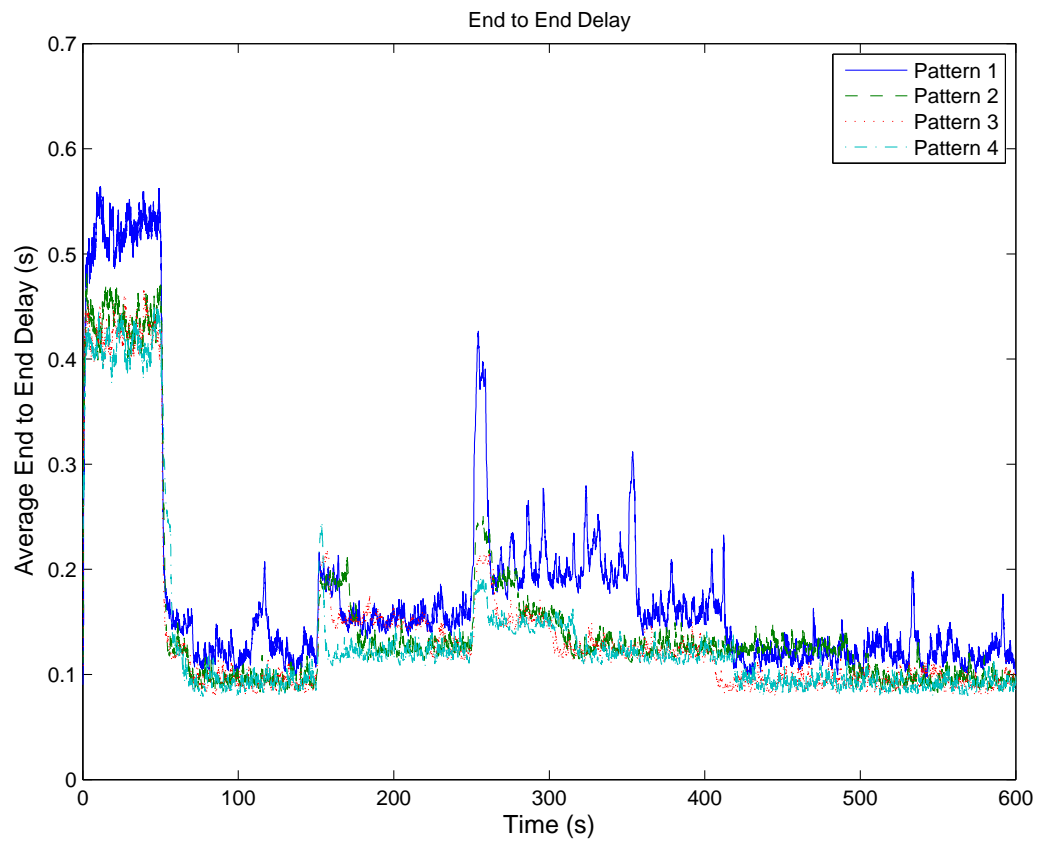


Figure 21: Average end to end delay using CEARS (Scenario 1)

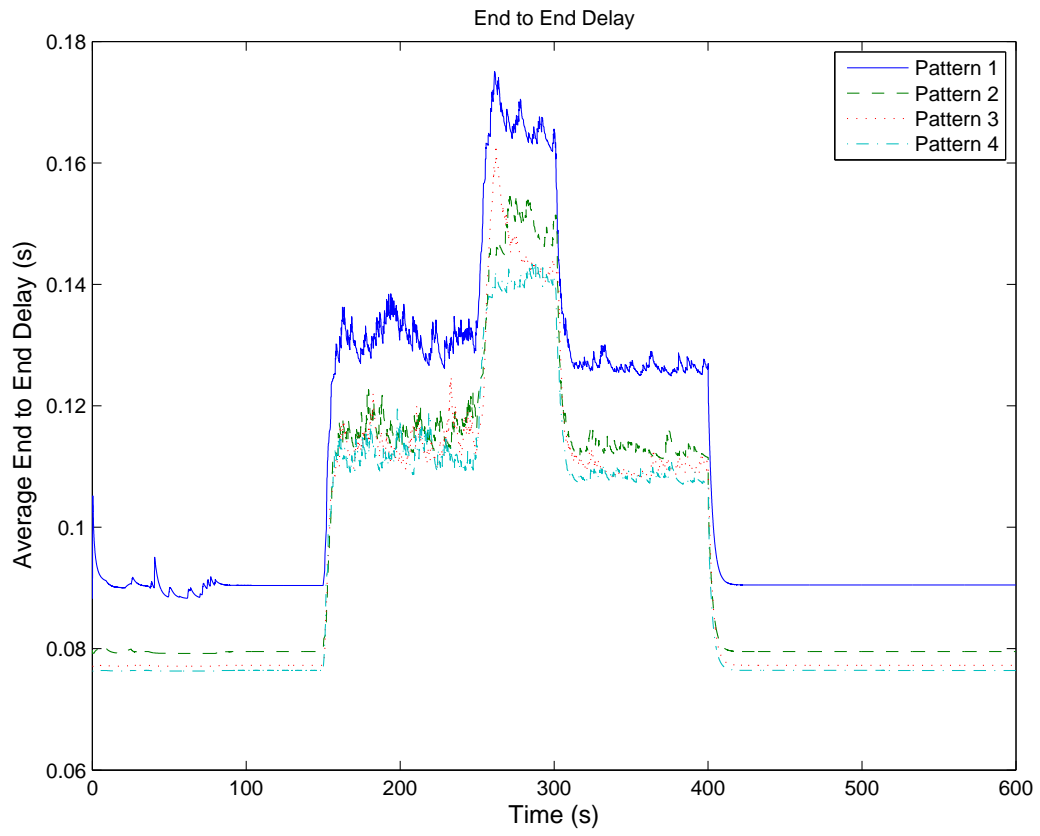


Figure 22: Average end to end delay using CEAARS (Scenario 1)

Table 4: Quality of paths found in simulation Scenario 1 Pattern 1 at different snapshots using the three different schemes. Values are obtained from the simulation after the first 11 snapshots (550 seconds) in the form of Mean(Standard Deviation) seconds

<i>Snap</i>	<i>Short Delay</i>	<i>CEARS Delay</i>	<i>CEAARS Delay</i>	<i>DV Delay</i>	<i>LS Delay</i>
$S_1$	0.0750	0.1208(0.0128)	0.0905(0.0000)	0.3541(0.0404)	0.4647(0.0754)
$S_2$	0.0750	0.1430(0.0561)	0.0899(0.0009)	0.3692(0.0023)	0.5010(0.0037)
$S_3$	0.0750	0.1292(0.0199)	0.0904(0.0000)	0.3743(0.0024)	0.5028(0.0029)
$S_4$	0.1050	0.1625(0.0197)	0.1302(0.0059)	0.3952(0.0414)	0.5296(0.0410)
$S_5$	0.1050	0.1557(0.0090)	0.1308(0.0020)	0.4108(0.0027)	0.5314(0.0070)
$S_6$	0.1350	0.2296(0.0611)	0.1649(0.0054)	0.4351(0.0419)	0.4336(0.0407)
$S_7$	0.1050	0.2057(0.0204)	0.1295(0.0071)	0.4503(0.0028)	0.4524(0.0040)
$S_8$	0.1050	0.1728(0.0379)	0.1263(0.0008)	0.4577(0.0025)	0.4553(0.0041)
$S_9$	0.0750	0.1314(0.0274)	0.0928(0.0065)	0.3542(0.0447)	0.4592(0.0172)
$S_{10}$	0.0750	0.1206(0.0092)	0.0905(0.0000)	0.3702(0.0023)	0.4654(0.0040)
$S_{11}$	0.0750	0.1246(0.0179)	0.0905(0.0000)	0.3750(0.0023)	0.4668(0.0028)

Table 5: Quality of paths found in simulation Scenario 1 Pattern 4 at different snapshots using the three different schemes. Values are obtained from the simulation after the first 11 snapshots (550 seconds) in the form of Mean(Standard Deviation) seconds

<i>Snap</i>	<i>Short Delay</i>	<i>CEARS Delay</i>	<i>CEAARS Delay</i>	<i>DV Delay</i>	<i>LS Delay</i>
$S_1$	0.0750	0.0909(0.0056)	0.0764(0.0000)	0.0777(0.0000)	0.0777(0.0000)
$S_2$	0.0750	0.1187(0.0607)	0.0763(0.0000)	0.0777(0.0000)	0.0777(0.0000)
$S_3$	0.0750	0.0932(0.0062)	0.0764(0.0000)	0.0777(0.0000)	0.0777(0.0000)
$S_4$	0.1050	0.1259(0.0233)	0.1108(0.0051)	0.1087(0.0003)	0.1088(0.0016)
$S_5$	0.1050	0.1219(0.0068)	0.1121(0.0020)	0.1087(0.0000)	0.1087(0.0006)
$S_6$	0.1350	0.1535(0.0134)	0.1391(0.0045)	0.1398(0.0003)	0.1398(0.0000)
$S_7$	0.1050	0.1295(0.0141)	0.1105(0.0059)	0.1093(0.0043)	0.1093(0.0042)
$S_8$	0.1050	0.1183(0.0054)	0.1082(0.0007)	0.1087(0.0000)	0.1087(0.0000)
$S_9$	0.0750	0.1034(0.0140)	0.0781(0.0049)	0.0783(0.0043)	0.0783(0.0043)
$S_{10}$	0.0750	0.0926(0.0065)	0.0764(0.0000)	0.0777(0.0000)	0.0777(0.0000)
$S_{11}$	0.0750	0.0907(0.0057)	0.0764(0.0000)	0.0777(0.0000)	0.0777(0.0000)

Table 6: Total routing overhead packets (packets) under Scenario 1

	<i>Pattern 1</i>	<i>Pattern 2</i>	<i>Pattern 3</i>	<i>Pattern 4</i>
<i>DV</i>	73874	74036	74034	74046
<i>LS</i>	343120	277266	342416	466161
<i>CEARS</i>	66085	66101	66194	66105
<i>CEAARS</i>	36649	37708	36758	36232

polar region. To clearly show the simulation results, we divide the network into 11 snapshots based on the status of interplane ISLs, and one snapshot lasts 50 seconds. Table 3 shows the interplane ISL's status, where  $S_i$  denotes the  $i^{th}$  snapshot;  $I_j$  denotes the  $j^{th}$  interplane; 0 at the  $i - th$  column and  $j^{th}$  row means that the interplane ISLs of the  $j^{th}$  interplane at the  $i^{th}$  snapshot are inactive, and 1 means active. In the simulation the source node is 3, and the destination node is 58. Maximum two hundred data packets with the size of 1000 bytes per packet are generated per second at the source node. For CEARS,  $\beta = 0.95$ , and 100 normal ants and 10 explorer ants are generated per second at the source node.

With the configuration explained above, CEAARS has been implemented and compared with the Distance Vector(DV) and Link State (LS) approaches, under both light and heavy load conditions as follows.

*Scenario 1 (Relation between load and performance):*

- *Pattern 1 (Heavy load):* the ISL bandwidth is set as  $1.5Mbit/s$ .
- *Pattern 2:* the ISL bandwidth is set as  $5Mbit/s$ .
- *Pattern 3:* the ISL bandwidth is set as  $10Mbit/s$ .
- *Pattern 4 (Light load):* the ISL bandwidth is set as  $15Mbit/s$ .

By inspection of the topology, we can calculate the minimum propagation delay from the source node to the destination node at the first and last three snapshots as  $75ms$ , at the  $4^{th}$ ,  $5^{th}$ ,  $7^{th}$  and  $8^{th}$  snapshots as  $105ms$ , and at the  $6^{th}$  snapshot as  $135ms$ . Figures 19, 20, 21 and 22 show the average end to end delay results in Scenario 1. We

can find that, at the first 50 seconds, the resulting delay using CEARS is around  $400ms$  which is quite high corresponding to the theoretical minimum delay at the two snapshots, while DV, LS, and CEAARS converge to the optimal paths very fast when the traffic load is light. DV and LS always select one of the shortest paths to forward packets. In the light load traffic situation, selecting one of the shortest paths to forward packets is the best selection. However, in the heavy load traffic situation, DV and LS cannot avoid the congested path. Table 4 and Table 5 show the statistical measures of the mean and standard deviation values in the extended simulation. By comparing Figs. 19, 20, and 22, also from Table 4, we can see that CEAARS outperforms DV and LS in heavy load situation, since CEAARS can adaptively select the minimum cost path. Moreover, from Table 6, we can find that the number of routing overhead packets in LS is much higher than in DV, CEARS and CEAARS, because the new state information in LS needs to be disseminated throughout the network by the LS flooding procedure. In DV algorithms, each node uses neighbors' routing tables to get the information of link costs to all its neighbors, and trigger update starts only when link changes are affecting the shortest path. This saves a lot of overhead packets especially in LEO satellite networks, where the ISLs become alternately inactive and active. In CEARS and CEAARS, backward ants update the routing table at intervals and there is no obligation to have global updating mechanisms. Overall, CEAARS demonstrates good adaptiveness to topology variations in simulations. Unlike DV and LS, CEAARS can adaptively choose the optimal path based on a cost function with no obligation to global updating mechanisms. Since CEAARS is capable of sensing the direction of destination, it remarkably improves the convergence speed. Since CEAARS can find the optimal or suboptimal paths quite soon at the beginning of the simulation, it is suitable for both short-time and long-time connections. The simulation on CEAARS also shows a significant saving in the number of ants with similar data packet delay and service availability. Consequently, CEAARS can be implemented as an on-demand routing system with low memory and computations.

## 2.4 Satellite Network TCP based Congestion Control using Convex Optimization

In addition to transmitting the SSA data across a single layer (such as LEO), the data also must often be transmitted between layers (such as to GEO). A multi-layered satellite network consisting of a large number of nano-satellites is ideally suited to perform space situational awareness. The large number of nano satellites collects visual information of space objects and transfer data to ground stations through a layered network in real time. Due to the dynamic topology of the network, large propagation delays and bulk data transfers, significant delays and loss of data can be encountered with traditional TCP congestion control schemes. To overcome these drawbacks, network snapshots consisting of layered dynamic clusters of satellites are formed with modified TCP congestion control schemes such that the delay and packet losses are at a minimum. The layered dynamic clusters are allowed to transmit data at maximum possible rates till congestion is detected. On congestion, an optimized data rate is determined using Convex Optimization techniques to support high data rates and reduce congestion. The multi-layered satellite architecture comprises of three layers. The lowest layer comprises of nano-satellites (N) with limited communication capabilities. Layer 2 comprises of MEO satellites (M) and layer 3 comprises of GEO-stationary satellites (G). Satellites communicate within layers via links known as the intra-orbital links whereas satellite communications between layers is done using the links known as inter-orbital links. Inter-orbital data flow is assumed to exist only in the lower and GEO layer. Only the GEO stationary satellites are assumed have the capabilities to communicate with the ground stations. The multi-layered satellite network is assumed so that the top layer will always represent the GEO layer and the other two layers can be located anywhere from the low earth orbits to right below GEO satellites. Next, the multi-layered architecture is further modified into a layered dynamic cluster of satellites by selecting network snapshots manager or cluster head satellites in each layer. First, several nano-satellites in the lowest layer

are chosen as the dynamic cluster heads based on certain communication capabilities and locations of neighboring nano-satellites. These cluster heads will collect data from its members in the lower layer and forward the data to a single satellite in the layer-2. Each satellite in the layer-2 will assume a role of dynamic cluster head however will communicate only with dynamic cluster heads in the lower layer and forward the received data to an appropriate GEO layer satellite. At the dynamic cluster heads in each layer the data is accumulated. Depending on the incoming data rate, size of the data buffers and the outgoing data rate, significant queuing delay would be introduced leading to drop in packets. A simple solution to avoid packet drops would be to have a large data buffer which is not practically feasible. Even if sufficient data buffers are available the queuing delay would still be significant due to low outgoing data rates. The other simple solution is to reduce significantly the incoming data rates which in turn would increase the end to end delay in transmitting the data to the ground station. Hence, there is a need for determining an optimal layered cluster and optimum incoming data rates. In this work, using convex optimization techniques, optimal incoming data rates are computed at each satellite irrespective of their layers. They are computed in each snapshot to keep the delay and packet loss at a minimum. During the process of determining optimal incoming data rates a collection of satellites are also identified to perform a particular space situational awareness task. In the next section, some of the results are presented and detailed discussions on the TCP modeling with convex optimization are provided in the appendix.

#### **2.4.1 Simulation Results**

A network comprising of 120 Nano-satellites in the lowest layer, 12 satellites in the middle layer and 3 GEO satellites were considered. It was assumed that only one of these GEO satellites has connectivity to the ground station located at Florida. The other two GEO satellites forward their data to the GEO satellite connected to the ground. This scenario was selected to stress the system and thereby demonstrate the



effect on buffer usage, bandwidth utilization, etc. The nano-satellites in the lower layer are assumed to have a transmission capability of one tenth that of the satellites in the middle layer. The GEO satellites are assumed to have the highest transmission capability. However, the physical distance (and in turn the propagation delay) between lower to middle layers and middle to GEO layers are significantly smaller than the propagation delays between GEO satellites and GEO satellite to the ground station. In a snapshot a GEO satellite could communicate with a maximum of four satellites in the middle layer and a satellite in the middle layer could communicate with a maximum of three cluster head satellites in the lower layer. The cluster size in the lower layer was restricted to five including the cluster head. Simulations were run for a fixed amount of time and performance parameters at each transmission were collected. The simulation run included a number of snapshots with fixed snapshot duration. It is assumed that there is no change in positions of the satellite during a snapshot. After a snapshot, satellites recalculate their positions, memberships/leadership and data rates. In Figures 23, 24, and 25, the average of the sum of buffer utilization of all cluster head satellites for each of the layers with snapshot duration of 120 secs and with traditional TCP congestion control are shown.

From Figure 23, it can be seen that the buffer utilization of the cluster head nano-satellites is around 20%. This is due to the small propagation delay between the lower and middle layer. The buffer utilization of the satellites in the middle layer is negligible due to the higher transmission rate capability between the middle layer and the GEO layer. However, in case of the GEO satellites as shown in Figure 25, the buffer utilization is greater than 100%. This is due to the large propagation delay between the GEO satellites and the ground station. This will introduce a significant delay and cause loss of packets. In Figure 26, the average of the sum of bandwidth utilization at GEO satellites is shown. In Figure 26, it can be seen that the bandwidth utilization consistently falls below 100% due to the TCP congestion control trying to adjust the window size through its built in slow start and congestion control mechanisms. In Figure 27, the average of the

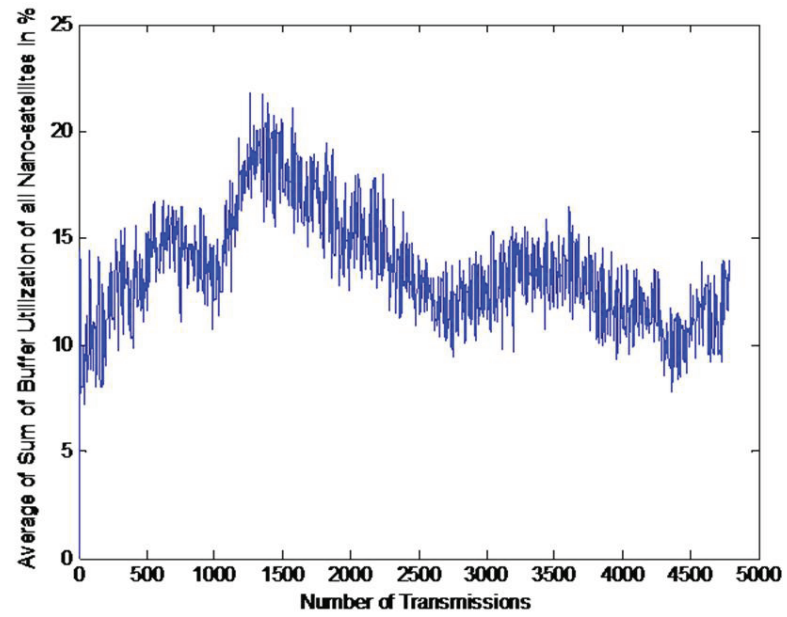


Figure 23: Average of Sum of Buffer Utilization of all cluster head nano-satellites

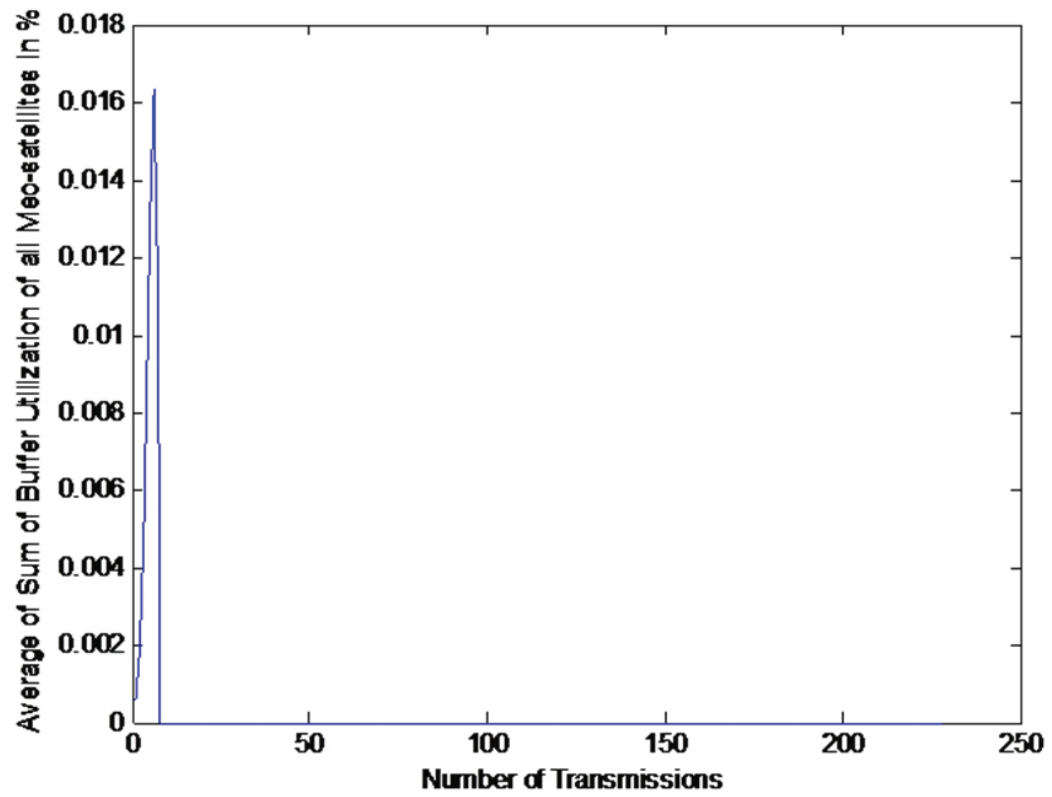


Figure 24: Average of Sum of Buffer Utilization of all satellites in the middle layer

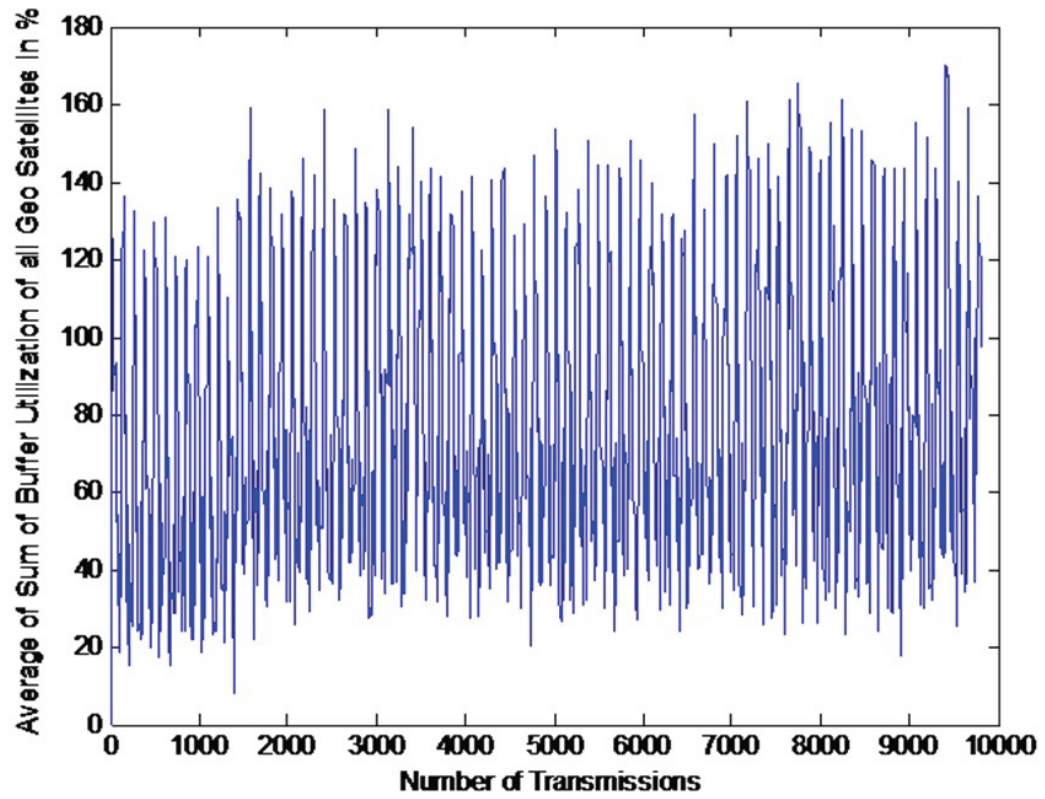


Figure 25: Average of Sum of Buffer Utilization of all satellites in the GEO layer

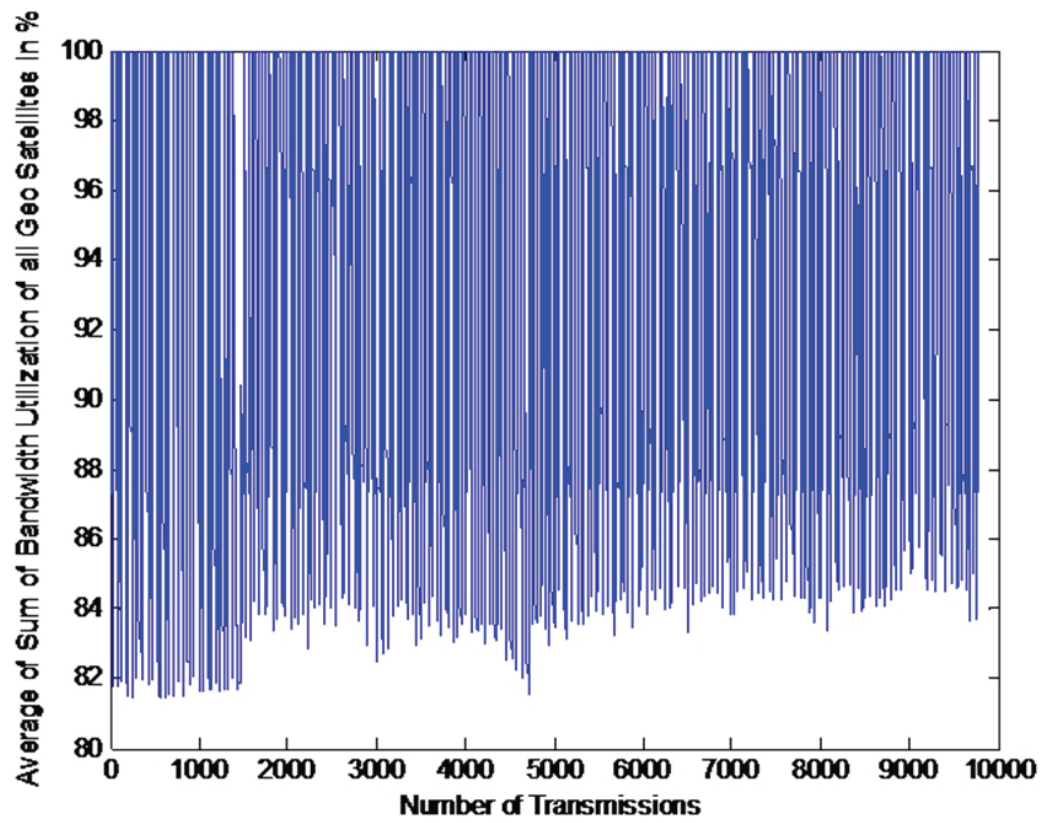


Figure 26: Average of Sum of Bandwidth Utilization of GEO satellites

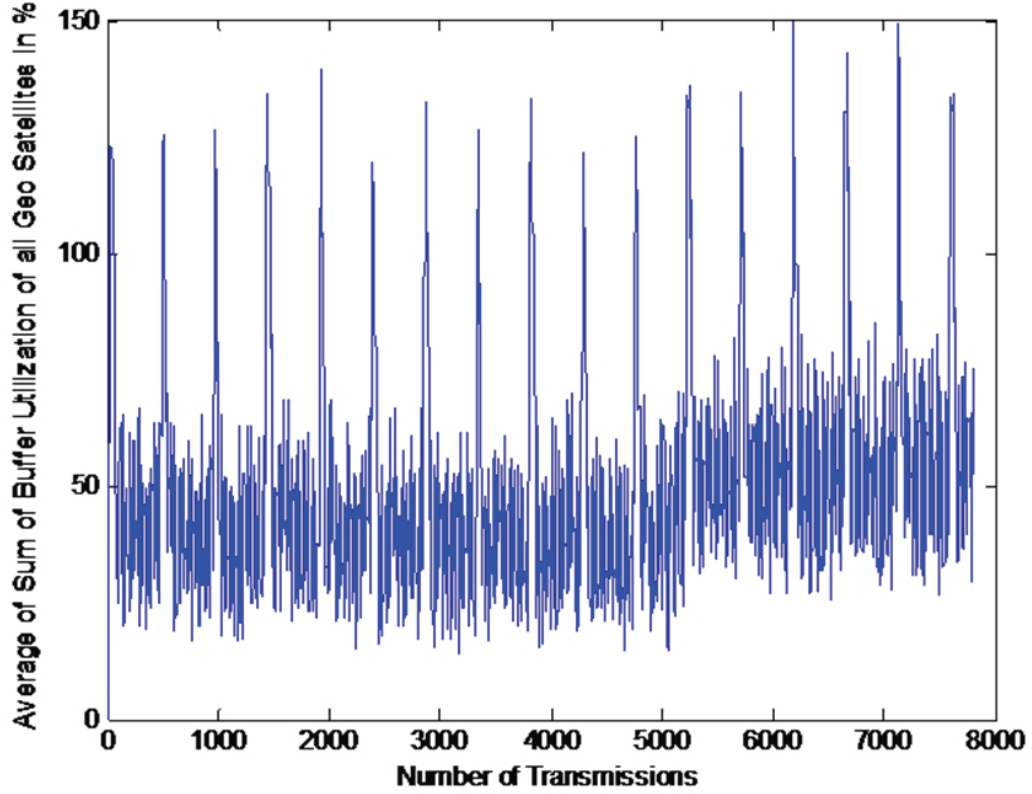


Figure 27: Average of Sum of Buffer Utilization of all satellites in the GEO layer

sum of buffer utilization of all GEO satellites with snapshot duration of 120 secs and with **convex optimization based TCP congestion control** is shown. Comparing Figures 25 and 27, it can be seen that the buffer utilization is always around 50% with the convex optimization based TCP congestion control. The spikes in buffer utilization are due to the satellites recalculating their positions, memberships/leadership and data rates at the beginning of a new snapshot. In a new snapshot at the beginning the traditional TCP congestion control is used. Once the congestion reaches a critical level, the convex optimization based control reduces the congestion. This can be seen in Figure 28, which shows the convex optimization mechanism operating in a single snapshot. In Figure 28, it can be seen that the buffer utilization is high

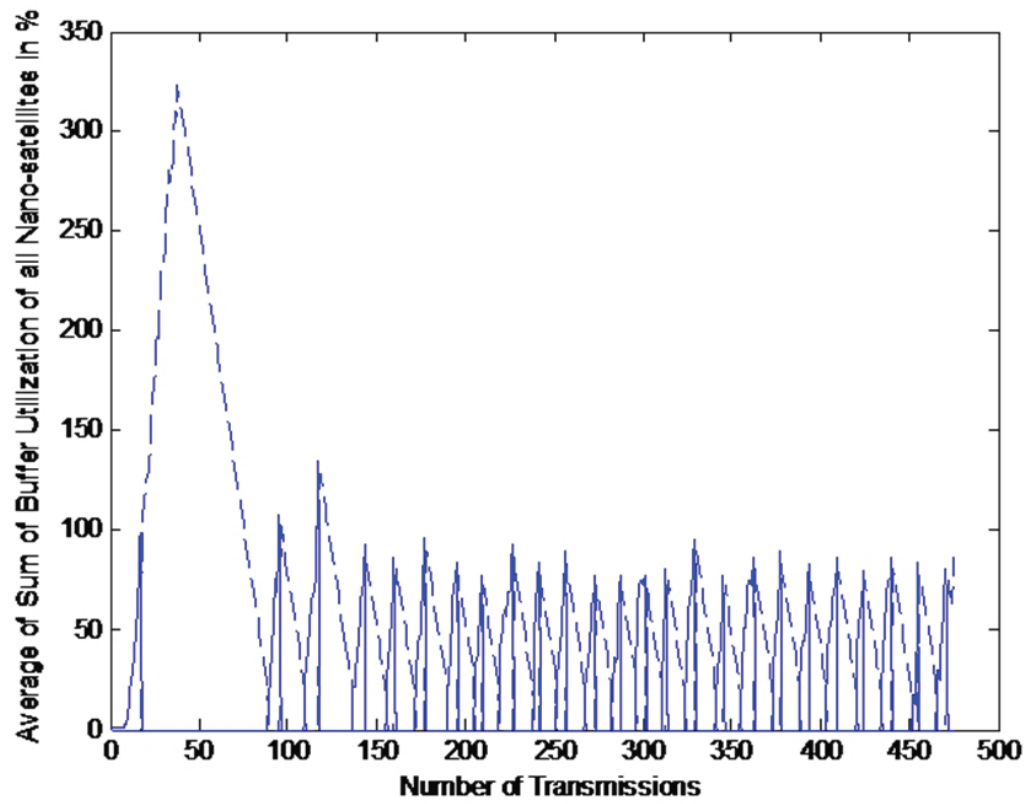


Figure 28: Buffer Utilization of GEO Satellites in a 120 second Snapshot

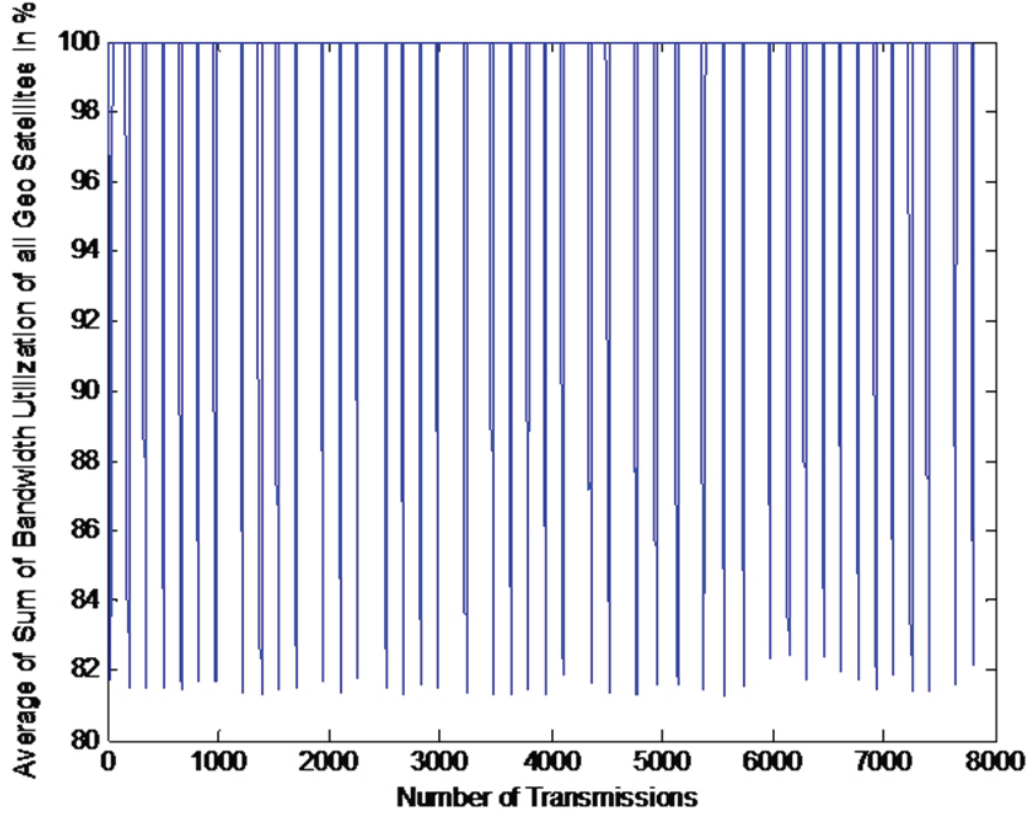


Figure 29: Average of Sum of Bandwidth Utilization of GEO satellites with Convex Optimization based TCP Congestion Control

at the beginning of the snapshot due to the traditional TCP congestion control. After the congestion reaches the critical level, the convex optimization based TCP congestion control kicks in and maintains the buffer utilization level around 50% for the entire duration of the snapshot. In Figure 29, the average of the sum of bandwidth utilization at GEO satellites with convex optimization based TCP congestion control is shown. Comparing Figures 27 and 29, it can be noticed that the GEO satellites with convex optimization based TCP congestion control ensures a constant 100 % bandwidth usage during a snapshot.



## 2.5 Models for Reconstruction of Images from Multiple Views

While most of this study concentrated on how SSA data can be collected and transmitted, we also studied basic signal processing methods suited for this SSA task. Reconstruction of detailed satellite images from multiple views and assembly of these views from orbital observations require one to ask: Under which transformations do band-limited signals/images remain invariant? This question was fully answered in our study sponsored by the AFOSR in [7]. In this paper, it is shown that these spaces remain invariant if and only if the transformations are affine. Equivalently stated, the class of warps preserving Paley-Wiener spaces are limited to translations and dilations. To account for aberrations and nonlinear warps, it was subsequently asked: What happens if the signal deformations are not affine? What are the function spaces to which the range signals/images belong and can one provide an inversion formula to recover the pre-images of these maps? To address this problem, structure theorems were constructed to realize images as members of certain reproducing kernel Hilbert spaces, where the reproducing kernels account for the aberration. In this orthogonal polynomial setting, best approximation to the signal/image is possible and an explicit inversion formula is derived. These results were also generalized from the Paley-Wiener spaces to the deBranges-Rovnyak spaces, which are the weighted analogues of the former spaces. The weights arise where illumination is not ideal and various obstructions are present. Statement of the exact theorems in [7] would introduce significant notational complexity. The referenced paper is intended to serve as a substitute.

Having determined the invariance properties of the image spaces and investigated best approximation approaches to dealing with the deformation and warping of these images, the question of feature tracking and realization of the warp(s) became essential. To describe the modeling, we shall use  $I : \mathbb{R}^3 \rightarrow \mathbb{R}^{2 \times 2}$  to denote the view of a 3-D RSO,  $\Omega \subset \mathbb{R}^3$ , by a camera. Depending on the model used for the camera, and whether optical, laser, RF or other methods are used,  $I$  can be appropriately adapted. If  $h : \Omega \times [0, \infty) \rightarrow \mathbb{R}^3$  is a time-dependent

diffeomorphism of  $\Omega$  into  $\mathbb{R}^3$  then a model for the deformation between two images of the same scene is given by

$$I_h(x, t) = I(h(x, t)) \quad \forall x \in \Omega \times \cap h^{-1}(\Omega) \subset \mathbb{R}^{2 \times 2}. \quad (28)$$

The image  $I_h(x, t)$  is the evolving image of the 3-D RSO, moving at a relatively high speed relative to a camera, as the RSO rotates and translates through space. If  $R(t) : [0, \infty) \in SO(3)$ , and  $T(t) : [0, \infty) \rightarrow \mathbb{R}^3$  relative to the Hill's frame, then  $h(x, t) = R(t)x + T(t)$  describes an affine deformation of the 3-D object. In this affine model, points in each image do not undergo the same motion but the motion of each point is related to its original location via an affine map. This model is a good approximation for small planar sections parallel to the image plane undergoing a rotation and translation about the optical axis, and small rotation about an axis orthogonal to the optical axis. More generally, if  $h(x, t) = H(x, t)x$ , where  $H$  is a  $3 \times 3$  matrix, any rigid body motion of a planar section in the scene can be captured. Since  $\Omega$  can be approximated by planar sections, this will lead to a satisfactory model except at discontinuities and occluding boundaries (relative to the normal to the image plane). Occlusions are represented by a real-valued factor  $m(x, t)$  multiplying the right hand side of eq. 28. The multiplier  $m$  may depend on the shape (e.g. curvature) of the surface, illumination of the surface, background lighting and possibly obstructions in the field of view. Clearly  $0 \leq m(x, t) \leq 1$ , where 1 implies that the point  $x$  on the surface is visible at time  $t$ , and 0 implies that it is not. Hence, eq. 28 can be recast in the form

$$I_{h,m}(x, t) = m(x, t)I(h(x, t)) \quad \forall x \in \Omega \times [0, \infty) \cap h^{-1}(\Omega) \subset \mathbb{R}^{2 \times 2}, \quad (29)$$

$$0 \leq m(x, t) \leq 1.$$

Since  $I_{h,m}(x, \cdot)$  must evolve continuously in time, it will be Markovian, and hence  $S_t(I(x, t)) := I_{h,m}(x, t) = m(x, t)I(h(x, t))$  is a one-parameter semigroup induced by the flow  $h : \Omega \times [0, \infty) \rightarrow \mathbb{R}^3$  and  $m$  is a cocycle for the flow  $h$ , i.e.  $m$  is an  $h$ -cocycle. Hence the image deformation and feature tracking within a given view can be stated as: Given a

time evolving scene,  $I_{h,m}$  induced by the flow  $h$  and h-cocycle  $m$ , determine  $h$  and  $m$ . This problem was addressed in the following paper [8]. In this paper, the class of all semigroups arising from cocycle-weighted time warps were characterized. Surprisingly, it was shown that the cocycle-weighted time warps give rise to very large classes of flows on Hardy spaces of analytic functions and, more importantly, a relatively simple criterion was developed to realize the weighted deformation as a cocycle-weighted time warp. This simple criterion is currently being specialized and applied to answering the following question: Which filters arise from time warps? [9].

Having established a correspondence between the multiple camera views, time evolving single camera views, and 3-D reconstruction of an RSO, feature detection from partial data is of utmost importance in orbit. Here the basic theory for processing the measurements also needs to be extended. Standard models of reconstruction of geometric features using multiple views utilizes data in each view and backprojects this data to construct the physical object. This method is extremely costly and, in the long run, may be quite inefficient in feature detection. Thus, it was asked: Is the entire data in each image necessary for a rapid and accurate reconstruction of features in an RSO? This question led us to investigate spectral methods for analyzing the camera images. Description of the 2-D views in appropriate bases that remains invariant under the image formation operator (i.e. the  $I$  operator) allows one to reduce the reconstruction significantly, while extracting the desired features. That is, we construct a family of orthonormal basis with respect an appropriate inner product, let  $T$  be an operator that maps the 2-D image into  $\ell^2$  with respect to this basis family, and require this basis to have the property that  $T$  commutes with the image operator  $I$ . For example, if  $T$  is the Fourier transform, it provides a representation in terms of the complex exponential basis, and  $T$  commutes with a slicing operator. Then the Fourier transform of a slice through a 3-D object is the same as the slice through the 3-D Fourier transform of that object. This is the projection-slice theorem. This theorem allows the transformation of the problem from a pointwise re-

construction of a geometric object from multiple views to a spectral reconstruction. Spectral reconstruction can be tailored to desired resolution and features. Determination of how the transform of a feature would look like in a view or a sequence of views can then be used, via correlation methods, to readily detect and identify those features. The results of this partial data reconstruction have been presented at a conference [10] and are being prepared for publication [11]. These results establish data spaces constructed from the partial data which converge in weak star topology to the complete data spaces. Realization of specific features in these data spaces, allow for detection of the desired features at much smaller cost. A closely related paper to this work, dealing with representation of the extreme points of the space of positive pluriharmonic function in Euclidean balls was also developed [12] under the AFOSR grant. The geometric tomography method developed this paper will have important application in the algorithm to be developed to describe complex images from multiple views.

### **3 List of Publications**

The reference section offers a complete list of the publications resulting from this project. The papers are organized as follows: (a) Papers ( [7], [13], [4], [12], [8]) are published in peer-reviewed journals ; (b) Papers ( [14], [15], [16], [17], [2], [1], [3], [10]) are published in non-peer-reviewed journals or in conference proceedings; (c) There are no papers presented at meetings, but not published in conference proceedings; (d) Papers ( [18], [19], [5], [20], [11], [9]) are manuscripts submitted or in preparation, but not published; and (e) Papers ( [21], [22], [23]) are M.S. and Ph.D. theses resulting from the grant.

### **4 Scientific Personnel Supported by this Project**

Raghuraj Prasanna Kumar (MS/PhD student); Suresh Muknahallipatna (faculty); Jinhua Cao (PhD student, graduated August 2010);

Dr. Margareta Stefanovic (faculty); Anup Gondkar (MS student, graduated December 2010); Amanda Johnson (MS student, graduated December 2008); Hamid Nourzadeh (PhD student, in progress); John McInroy (faculty); Farhad Jafari (faculty).

## 5 Report of Inventions

None.

## References

- [1] J. Cao and M. Stefanovic, “New results in stable switching congestion control for satellite TCP/AQM networks,” in *In Proc. of 49th IEEE Control and Decision Conference*, Atlanta, GA, December 2010.
- [2] J. Cao and M. Stefanovic, “Switching congestion control for satellite TCP/AQM networks,” in *American Control Conference*, (St. Louis, MO), June 2009.
- [3] J. Cao and M. Stefanovic, “Cross entropy accelerated ant routing in satellite networks,” in *In Proc. of American Control Conference*, Baltimore, MD, July 2010.
- [4] J. Cao and M. Stefanovic, “Cross entropy accelerated ant routing in satellite network: Comparison with a benchmark routing system,” *Swarm Intelligence Journal*, provisionally accepted, 2010.
- [5] M. Stefanovic and J. Cao, “Stability and performance guarantees for congestion control of satellite TCP/AQM networks using unfalsified control,” *Submitted to International Journal of Adaptive Control and Signal Processing*, 2010.
- [6] M. Stefanovic and M. G. Safonov, “Safe adaptive switching control: Stability and convergence,” *IEEE Trans. Autom. Control*, vol. 53, pp. 2012–2021, Jan 2008.

- [7] S. Mukherjee, F. Jafari, and J. McInroy, “On characterization of range spaces of composition operator on spaces of entire functions,” *To Appear, Contemporary Mathematics AMS*, 2011.
- [8] F. Jafari, Z. Slodkowski, and T. Tonev, “Semigroups of operators on hardy spaces and cocycles of holomorphic flows,” *Complex Analysis and Operator Theory*, published online 16 April 2010.
- [9] F. Jafari, S. Mukherjee, and J. McInroy, “Filters arising from time warps,” *In preparation*.
- [10] F. Jafari, “Variants on a classical moment problem,” in *6th International Conference on Function Spaces*, (Edwardsville, IL), May 2010.
- [11] F. Jafari, J. McInroy, and D. Cochran, “Image reconstruction from partial data,” *In preparation*.
- [12] F. Jafari and M. Putinar, “Extremal positive pluriharmonic functions on euclidean balls,” *Pure and Applied Math Quarterly*, vol. 6, no. 4, pp. 1013–1027, 2010.
- [13] J. E. McInroy, L. M. Robertson, and R. S. Erwin, “Autonomous distant visual silhouetting of satellites,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 801–808, 2008.
- [14] R. Buffington and J. E. McInroy, “Homography-based change detection for space based satellite inspection,” in *SPIE Conference on Defense, Security, and Sensing*, (Orlando, FL), April 2011.
- [15] R. Buffington and J. E. McInroy, “Change detection for visual satellite inspection using pose estimation and image synthesis,” in *SPIE Conference on Defense, Security, and Sensing*, (Orlando, FL), April 2011.
- [16] H. Nourzadeh and J. E. McInroy, “Planning the visual measurement of  $n$  moving objects by  $m$  moving cameras, given their relative trajectories,” in *IEEE International Conference on Computer Vision and Pattern Recognition*, (Colorado Springs, CO), June 2011.

- [17] J. E. McInroy and L. Robertson, “Optimal observation of satellites using combined measurements from many networked observers cooperatively controlled by human and autonomous means,” in *AIAA Conference on Guidance, Control, and Navigation*, (Toronto, Canada), August 2010.
- [18] F. Jafari and J. McInroy, “On interpolation methods in image reconstruction,” *in preparation (to be submitted to IEEE Transaction on Visualization and CG)*, 2010.
- [19] R. P. Kumar, S. Muknahallipatna, M. Stefanovic, and J. McInroy, “Satellite network TCP based congestion control using convex optimization,” *EURASIP Journal on Wireless Communications and Networking*, To be submitted, July 2011.
- [20] M. Stefanovic, S. Muknahallipatna, and J. Cao, “Congestion control and performance analysis of the TCP networks using OP-NET,” *Journal of Networks and Computer Applications*, to be submitted, 2011.
- [21] A. R. Johnson, “A pose estimation algorithm based on points to regions correspondence using multiple viewpoints,” MS thesis, University of Wyoming, December 2008.
- [22] A. Gondkar, “Optimal observation of satellites using combined measurements from many networked observers,” MS thesis, University of Wyoming, December 2010.
- [23] J. Cao, *Congestion Control and Routing over Satellite Networks*. PhD thesis, University of Wyoming, April 2010.